



Setoid type theory - a syntactic translation

Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, Nicolas Tabareau

► To cite this version:

Thorsten Altenkirch, Simon Boulier, Ambrus Kaposi, Nicolas Tabareau. Setoid type theory - a syntactic translation. MPC 2019 - 13th International Conference on Mathematics of Program Construction, Oct 2019, Porto, Portugal. pp.155-196, 10.1007/978-3-030-33636-3_7 . hal-02281225

HAL Id: hal-02281225

<https://inria.hal.science/hal-02281225>

Submitted on 8 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Setoid type theory — a syntactic translation[★]

Thorsten Altenkirch¹, Simon Boulier², Ambrus Kaposi³, and
Nicolas Tabareau²

¹ University of Nottingham, UK

`Thorsten.Altenkirch@nottingham.ac.uk`

² Inria, Nantes, France

`{simon.boulier,nicolas.tabareau}@inria.fr`

³ Eötvös Loránd University, Budapest, Hungary
`akaposi@inf.elte.hu`

Abstract. We introduce setoid type theory, an intensional type theory with a proof-irrelevant universe of propositions and an equality type satisfying function extensionality, propositional extensionality and a definitional computation rule for transport. We justify the rules of setoid type theory by a syntactic translation into a pure type theory with a universe of propositions. We conjecture that our syntax is complete with regards to this translation.

Keywords: type theory · function extensionality · proof irrelevance · univalence

1 Introduction

Extensional type theory (ETT [23]) is a convenient setting for formalising mathematics: equality reflection allows replacing provably equal objects with each other without the need for any clutter. On paper this works well, however computer checking ETT preterms is hard because they don’t contain enough information to reconstruct their derivation. From Hofmann [16] and later work [26,32] we know that any ETT derivation can be rewritten in intensional type theory (ITT) extended with two axioms: function extensionality and uniqueness of identity proofs (UIP). ITT preterms contain enough information to allow computer checking, but the extra axioms⁴ introduce an inconvenience: they prevent certain computations. The axioms act like new neutral terms which even appear in the empty context: a boolean in the empty context is now either true or false or a neutral term coming from the axiom. This is a practical problem: for computer formalisation one main advantage of type theory over to set theory is that

[★] This work was supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002), by COST Action EUTypes CA15123, by ERC Starting Grant CoqHoTT 637339, EPSRC grant EP/M016994/1 and by USAF, Airforce office for scientific research, award FA9550-16-1-0029 and by the National Research, Development and Innovation Office – NKFIH project 127740.

⁴ The problem is only with the axiom of function extensionality as adding UIP using Streicher’s axiom K [29] doesn’t pose a challenge to normalisation.

certain equalities are trivially true by computation, and the additional axioms limit this computational power.

In general, the usage of an axiom is justified by a model [18] in which the axiom holds. For example, the cubical set model [8] justifies the univalence axiom, the reflexive graph model [6] justifies parametricity, the groupoid model [19] justifies the negation of UIP, the setoid model [17,1] justifies function extensionality. A model can help designing a new type theory in which the axiom holds and which has full computational power, i.e. normalisation. Examples are cubical type theory [12] inspired by the cubical set model [8] and observational type theory [4] inspired by the setoid model [1].

In this paper we revisit the problem of designing a type theory based on the setoid model. We derive *setoid type theory* from the setoid model using an intermediate *syntactic translation*.

Most models interpret syntactic objects by metatheoretic structures, usually the ones they are named after. In the cubical model, a context (or a closed type) is a cubical set, in the groupoid model a context is a groupoid, and so on. Syntactic models [10] are special kinds of models: they interpret syntax by the syntax of another (or the same) theory. We call the interpretation function into such a model a *syntactic translation*. Equal (convertible) terms are equal objects in a model, which means that convertible terms are translated to convertible terms in the case of a syntactic model. This restricts the number of models that can be turned into syntactic models. A sufficient (but not necessary) criterion to turn a model into a syntactic model is the *strictness* of the model which means that all the equality proofs in the model are given by reflexivity (i.e. they are definitional equalities of the metatheory). Giving the metatheory an explicit syntax and renaming it target theory, a strict model can be turned into a syntactic translation from the source theory to the target theory. We will give examples of this process later on.

The setoid model given by Altenkirch [1] is a strict model, hence it can be phrased as a syntactic translation. A closed type in the setoid model is a set together with an equivalence relation. There are several ways to turn this model into a syntactic model, but in one of these a closed type is given by (1) a type, (2) a binary relation on terms of that type and (3) terms expressing that the relation (2) is reflexive, symmetric and transitive. We will define the syntax for setoid type theory by reifying parts of this model: we add the definitions of the relation (2) and its properties (3) as new term formers to type theory. The new equality type (identity type) will be the relation (2). The equalities describing the translation will be turned into new definitional equality rules of the syntax. Thus the new equality type will satisfy function extensionality and propositional extensionality by definition.

We also extend the setoid translation with a new rule making the elimination principle of equality compute definitionally.

In this paper we do not aim to give a precise definition of the notion of syntactic model or the relationship between different kinds of models. Our main goal is to obtain a convenient syntax for setoid type theory.

Structure of the paper. After summarising related work, in Section 2 we introduce $\text{MLTT}_{\text{Prop}}$, Martin-Löf type theory extended with a definitionally proof irrelevant universe of propositions [14]. In Section 3, we illustrate how to turn models into syntactic translations by the examples of the standard (set) model and the graph model. One of the syntactic translation variants of the graph model turns out to be Bernardy et al’s parametricity translation [7]. The model corresponding to this translation is not strict, showing that strictness is not a necessary condition for a model to have a syntactic variant. In Section 4 we define the setoid model as a syntactic translation. We also show that this translation can be extended with a new component saying that transport (the eliminator of equality) computes definitionally. In Section 5, we reflect the setoid translation into the syntax of $\text{MLTT}_{\text{Prop}}$ obtaining a new definition of a heterogeneous equality type. We also show that the translation of Section 4 extends to this new equality type and we compare it to the old-style inductive definition of equality. We conclude in Section 6.

Contributions. Our main contribution is the new heterogeneous equality type which, as opposed to John Major equality [4], is not limited to proof-irrelevant equality and is much simpler than cubical equality types [12,28]. As opposed to [4,28] we do not need to go through extensional type theory to justify our syntax but we do this by a direct translation into a pure intensional type theory. In addition to function extensionality, our setoid type theory supports propositional extensionality and a definitional computation rule for transport, which is also a new addition to the setoid model. The results were formalised in Agda.

Formalisation. The model variant ($|-_0$ variant in Section 3) of the setoid translation has been formalised [21] in Agda using the built-in definitionally proof irrelevant Prop universe of Agda. The formalisation includes the definitional computation rule for transport and does not use any axioms. In addition to what is described in this paper, we show that this model supports quotient types and universes of sets where equality is given by equality of codes.

1.1 Related work

A general description of syntactic translations for type theory is given in [10]. In contrast with this work, our translations are defined on intrinsic (well-typed) terms. A translation inspired by [7] for deriving computation rules from univalence is given in [30]. This work does not define a new type theory but recovers some computational power lost by adding the univalence axiom. A syntactic translation for the presheaf model is given in [20].

The setoid model was first described by [17] in order to add extensionality principles to type theory such as function extensionality and equality of logically equivalent propositions. A strict variant of the setoid model was given by [1] using a definitionally proof-irrelevant universe of propositions. Recently, support for such a universe was added to Agda and Coq [14] allowing a full formalisation of Altenkirch’s setoid model. Observational type theory (OTT) [4] is a syntax

for the setoid model differing from our setoid type theory by using a different notion of heterogeneous equality type, McBride’s John Major equality [25]. We show the consistency of our theory using the setoid translation, while OTT is translated to extensional type theory for this purpose [4]. XTT [28] is a cubical variant of observational type theory where the equality type is defined using an interval pretype. Supporting this pretype needs much more infrastructure than our new rules for setoid type theory.

A very powerful extensionality principle is Voevodsky’s univalence axiom [27]. The cubical set model of type theory [8] is a constructive model justifying this axiom. A type theory extracted from this model is cubical type theory [12]. The relationship between the cubical set model and cubical type theory is similar to the relationship between the setoid model and setoid type theory.

Previously we attempted to use a heterogeneous equality type similar to the one coming from the setoid translation to define a cubical type theory [3]. This work however is unfinished: the combinatorial complexity arising from equalities between equalities so far prevents us from writing down all the computation rules for that theory. In the setoid case, this blow up is avoided by forcing the equality to be a proposition.

Compared to cubical type theories [12,28], our setoid type theory has the advantage that the equality type satisfies more definitional equalities: while in cubical type theory equality of pairs is isomorphic⁵ to the pointwise equalities of the first and second components, in our case the isomorphism is replaced by a definitional equality. The situation is similar for other type formers. These additional definitional equalities are the main motivation for Herbelin’s proposal for a cubical type theory [15]. As setoid type theory supports UIP (Streicher’s axiom K, [29]), it is incompatible with full univalence. The universe of propositions in setoid type theory satisfies propositional extensionality, which is the version of univalence for mere propositions. However, this is not a subobject classifier in the sense of Topos Theory since it doesn’t classify propositions in the sense of HoTT (it seems to be a quasi topos though).

Setoid type theory is not homotopy type theory restricted to homotopy level 0 (the level of sets, or h-sets). This is because the universe of propositions we have is static: we don’t have that for any type, if any two elements of it are equal, then it is a proposition. The situation is similar for the groupoid model [19] which features a static universe of sets (h-sets).

2 MLTT_{Prop}

MLTT_{Prop} is an intensional Martin-Löf type theory with Π , Σ , Bool types and a static universe of strict propositions. We present MLTT_{Prop} using an algebraic (intrinsic) syntax [2], that is, there are only well-typed terms so preterms or typing relations are never mentioned. Conversion (definitional equality) rules are

⁵ This is a definitional isomorphism: A and B are definitionally isomorphic, if there is an $f : A \rightarrow B$, a $g : B \rightarrow A$ and $\lambda x.f(gx) = \lambda x.x$ and vice versa where $=$ is definitional equality.

given by equality constructors (using the metatheoretic equality $=$), so the whole syntax is quotiented by conversion. As a consequence, all of the constructions in this paper have to preserve typing and definitional equality. In this section we explain the syntax for this type theory listing the most important rules. The full signature of the algebraic theory $\text{MLTT}_{\text{Prop}}$ is given in Appendix A.

There are four sorts: contexts, types, substitutions and terms. Contexts and types are stratified into separate (predicative, cumulative) levels, as indicated by the indices i, j . In the Agda formalisation we use explicit lifting operations instead of cumulativity.

$$\frac{}{\text{Con}_i : \text{Set}} \quad \frac{\Gamma : \text{Con}_i}{\text{Ty}_j \Gamma : \text{Set}} \quad \frac{\Gamma : \text{Con}_i \quad \Delta : \text{Con}_j}{\text{Sub } \Gamma \Delta : \text{Set}} \quad \frac{\Gamma : \text{Con}_i \quad A : \text{Ty}_j \Gamma}{\text{Tm } \Gamma A : \text{Set}}$$

We use the following naming conventions for metavariables: universe levels i, j ; contexts $\Gamma, \Delta, \Theta, \Omega$; types A, B, C ; terms t, u, v, w, a, b, c, e ; substitutions δ, ν, τ, ρ . Constructors of the syntax are written in **red** to help distinguish from definitions. Most constructors have implicit arguments, e.g. type substitution below $-[-]$ takes the two contexts Γ and Δ as implicit arguments.

The syntax for the substitution calculus is the following. It can also be seen as an unfolding of category with families (CwF, [13]) with the difference that we write variable names and implicit weakenings instead of De Bruijn indices.

$$\begin{array}{c} \frac{}{\cdot : \text{Con}_0} \quad \frac{\Gamma : \text{Con}_i \quad A : \text{Ty}_j \Gamma}{(\Gamma, x : A) : \text{Con}_{i \sqcup j}} \quad \frac{A : \text{Ty}_i \Delta \quad \delta : \text{Sub } \Gamma \Delta}{A[\delta] : \text{Ty}_i \Gamma} \\[10pt] \frac{\Gamma : \text{Con}_i}{\text{id}_\Gamma : \text{Sub } \Gamma \Gamma} \quad \frac{\delta : \text{Sub } \Theta \Delta \quad \nu : \text{Sub } \Gamma \Theta}{\delta \circ \nu : \text{Sub } \Gamma \Delta} \quad \frac{\Gamma : \text{Con}_i}{\epsilon_\Gamma : \text{Sub } \Gamma \cdot} \\[10pt] \frac{\delta : \text{Sub } \Gamma \Delta \quad t : \text{Tm } \Gamma (A[\delta])}{(\delta, x \mapsto t) : \text{Sub } \Gamma (\Delta, x : A)} \quad \frac{\delta : \text{Sub } \Gamma (\Delta, x : A)}{\delta : \text{Sub } \Gamma \Delta} \quad \frac{\delta : \text{Sub } \Gamma (\Delta, x : A)}{x[\delta] : \text{Tm } \Gamma (A[\delta])} \\[10pt] \frac{t : \text{Tm } \Delta A \quad \delta : \text{Sub } \Gamma \Delta}{t[\delta] : \text{Tm } \Gamma (A[\delta])} \quad [\text{id}] : A[\text{id}] = A \quad [\circ] : A[\delta \circ \nu] = A[\delta][\nu] \\[10pt] \text{id} \circ : \text{id} \circ \delta = \delta \quad \circ \text{id} : \delta \circ \text{id} = \delta \quad \circ \circ : (\delta \circ \nu) \circ \tau = \delta \circ (\nu \circ \tau) \\[10pt] \cdot \eta : (\delta : \text{Sub } \Gamma \cdot) = \epsilon \quad , \beta_0 : (\delta, x \mapsto t) = \delta \quad , \beta_1 : x[(\delta, x \mapsto t)] = t \\[10pt] , \eta : (\delta, x \mapsto x[\delta]) = \delta \quad , \circ : (\delta, x \mapsto t) \circ \nu = (\delta \circ \nu, x \mapsto t[\nu]) \end{array}$$

There are two ways of forming a context: the empty context and context extension (or comprehension; here $i \sqcup j$ denotes the maximum of i and j). In context extension, the $:$ after the variable x is just notation, it differs from the metatheoretic colon. A substitution $\text{Sub } \Gamma \Delta$ can be thought of as a list of terms, one for each type in Δ , all given in context Γ . Such a substitution δ acts on a type $A : \text{Ty}_i \Delta$ by $A[\delta] : \text{Ty}_i \Gamma$. Note that $-[-]$ is a constructor, not an operation, that is, we are defining an explicit substitution calculus. There are five ways to form substitutions: identity **id**, composition \circ , the empty substitution ϵ , extending

a substitution with a term and forgetting the last term in the substitution (this is an implicit constructor). Terms can be formed using a variable (projecting out the last component from a substitution) and by action of a substitution.

We use variable names for readability, however these should be understood formally as the well-typed De Bruijn indices of CwFs, hence we consider α -equivalent terms equal. In the formalisation we use De Bruijn indices. We denote variables by $x, y, z, f, \gamma, \alpha$.

The equalities of the substitution calculus can be summarised as follows: type substitution is functorial, contexts and substitutions form a category with a terminal object \cdot and substitutions $\text{Sub } \Gamma(\Delta, x : A)$ are in a natural one-to-one correspondence with substitutions $\delta : \text{Sub } \Gamma \Delta$ and terms $\text{Tm } \Gamma(A[\delta])$. Ordinary variables can be recovered by $x := x[\text{id}]$. Weakenings are implicit.

In equation $\cdot\eta$, a type annotation is added on δ to show that this equation is only valid for δ s with codomain \cdot . Implicit weakenings are present in equations $\cdot\beta_0$ and $\cdot\eta$. Note that equation $\cdot\circ$ is only well-typed because of a previous equation: $t[\nu]$ has type $A[\delta][\nu]$, but it needs type $A[\delta \circ \nu]$ to be used in an extended substitution. In our informal notation, we use extensional type theory [23] as metatheory, hence we do not write such transports explicitly.⁶ However all of our constructions can be translated to an intensional metatheory with function extensionality and uniqueness of identity proofs (UIP) following [16,26,32].

We sometimes omit the arguments written in subscript as e.g. we write id instead of id_F . We write $t[x \mapsto u]$ for $t[(\text{id}, x \mapsto u)]$.

Dependent function space is given by the following syntax.

$$\frac{A : \text{Ty}_i \Gamma \quad B : \text{Ty}_j(\Gamma, x : A)}{\Pi(x : A).B : \text{Ty}_{i \sqcup j} \Gamma} \quad \frac{t : \text{Tm}(\Gamma, x : A) B}{\lambda x. t : \text{Tm} \Gamma(\Pi(x : A).B)}$$

$$\frac{t : \text{Tm} \Gamma(\Pi(x : A).B)}{t @ x : \text{Tm}(\Gamma, x : A) B} \quad \Pi\beta : (\lambda x. t) @ x = t \quad \Pi\eta : \lambda x. t @ x = t$$

$$\Pi[] : (\Pi(x : A).B)[\nu] = \Pi(x : A[\nu]).B[\nu] \quad \lambda[] : (\lambda x. t)[\nu] = \lambda x. t[\nu]$$

We write $A \Rightarrow B$ for $\Pi(x : A).B$ when x does not appear in B . The usual application can be recovered from the categorical application $@$ using a substitution and we use the same $@$ notation: $t @ u := (t @ x)[x \mapsto u]$. $\Pi[]$ and $\lambda[]$ are the substitution laws for Π and λ , respectively. A substitution law for $@$ can be derived using $\lambda[], \Pi\beta$ and $\Pi\eta$.

The syntax of dependent pairs and booleans follows the same principles and is given in Appendix A for the completeness of the presentation.

We have a hierarchy of universes of strict propositions. Any two elements of a proposition are definitionally equal: this is expressed by the rule irr_a (recall

⁶ Note that this does not mean that when defining our setoid model we rely on extensionality of the metatheory: our models will be given as syntactic translations as described in Section 3.

that $=$ is the equality of the metatheory).

$$\frac{}{\text{Prop}_i : \text{Ty}_{i+1} \Gamma} \quad \frac{a : \text{Tm } \Gamma \text{ Prop}_i}{\underline{a} : \text{Ty}_i \Gamma} \quad \frac{u : \text{Tm } \Gamma \underline{a} \quad v : \text{Tm } \Gamma \underline{a}}{\text{irr}_a : u = v}$$

This universe is closed under dependent function space, dependent sum, unit and empty types. Decoding an element of **Prop** is written using underline instead of the usual **El**. We use a, b, c as metavariables of type **Prop** and lowercase π and σ for the proposition constructors. The domain of the function space needs not be a proposition however needs to have the same universe level. For π and σ the constructors and destructors are overloaded. We also write \Rightarrow and \times for the non-dependent versions of π and σ . The syntax is given below (for the substitution laws see Appendix A).

$$\begin{array}{c} \frac{A : \text{Ty}_i \Gamma \quad b : \text{Tm } (\Gamma, x : A) \text{ Prop}_j}{\pi(x : A).b : \text{Tm } \Gamma \text{ Prop}_{i \sqcup j}} \quad \frac{t : \text{Tm } (\Gamma, x : A) \underline{b}}{\lambda x. t : \text{Tm } \Gamma \underline{\pi(x : A).b}} \\[10pt] \frac{t : \text{Tm } \Gamma \underline{\pi(x : A).b}}{t @ x : \text{Tm } (\Gamma, x : A) \underline{b}} \quad \frac{a : \text{Tm } \Gamma \text{ Prop}_i \quad b : \text{Tm } (\Gamma, x : \underline{a}) \text{ Prop}_j}{\sigma(x : a).b : \text{Tm } \Gamma \text{ Prop}_{i \sqcup j}} \\[10pt] \frac{u : \text{Tm } \Gamma \underline{a} \quad v : \text{Tm } \Gamma \underline{b[x \mapsto u]}}{(u, v) : \text{Tm } \Gamma \underline{\sigma(x : a).b}} \quad \frac{t : \text{Tm } \Gamma \underline{\sigma(x : a).b}}{\text{pr}_0 t : \text{Tm } \Gamma \underline{a}} \quad \frac{t : \text{Tm } \Gamma \underline{\sigma(x : a).b}}{\text{pr}_1 t : \text{Tm } \Gamma \underline{a[x \mapsto \text{pr}_0 t]}} \\[10pt] \frac{}{\top : \text{Tm } \Gamma \text{ Prop}_0} \quad \frac{}{\text{tt} : \text{Tm } \Gamma \underline{\top}} \quad \frac{}{\perp : \text{Tm } \Gamma \text{ Prop}_0} \quad \frac{C : \text{Ty}_i \Gamma \quad t : \text{Tm } \Gamma \underline{\perp}}{\text{exfalse } t : \text{Tm } \Gamma C} \end{array}$$

Note that we do not need to state definitional equalities of proofs of propositions such as β for function space, as they are true by **irr**. Definitional proof-irrelevance also has the consequence that for any two pairs (t, u) and (t', u') which both have type $\underline{\Sigma(x : A).b}$, whenever $t = t'$ we have $(t, u) = (t', u')$. We will use this fact later.

3 From model to translation

In this section, as a warm-up for the setoid translation, we illustrate the differences between models and syntactic translations by defining three different syntactic translation variants of the standard model (Subsection 3.1) and then showing what the corresponding translations for the graph model are (Subsection 3.2). One of these will be the parametricity translation of Bernardy et al [7].

A *model* of $\text{MLTT}_{\text{Prop}}$ is an algebra of the signature given in Section 2 and fully in Appendix A. In categorical terms, a model is a CwF with extra structure but informally expressed with named variables. The syntax is the initial model which means that for every model M there is a unique interpretation function from the syntax to M (usually called the recursor or eliminator). Below we define models by their interpretation functions: we first provide the specification of the

functions (what contexts, types, substitutions and terms are mapped to), then provide the implementation of the functions by listing how they act on each constructor of the syntax. This includes the equality constructors (conversion rules) that the interpretation needs to preserve. For a formal definition of how to derive the notion of model, interpretation function and related notions from a signature, see [22].

As opposed to the notion of model, the notion *syntactic model* (or its interpretation function, *syntactic translation*) is informal. Contexts in a model are usually given by some metatheoretic structure (e.g. sets, graphs, cubical sets, setoids, etc.) and similarly for types, substitutions and terms. In a syntactic model, contexts are given by syntax of another type theory called the *target theory* (this syntax could be contexts of the target theory, terms of the target theory, or a combination of both and also some equalities, etc.). We will illustrate the possibilities with several examples below. It is usually harder to define syntactic models than models, because of the equalities (conversion rules) the model has to satisfy. In a model these equalities are propositional equalities of the metatheory, while in a syntactic model equalities are definitional equalities (conversion rules) of the target theory. A basic example to illustrate this difference is given by extensional type theory (ETT). ETT has a *model* in an intensional metatheory with function extensionality (the standard interpretation $|-|_0$ below works: equality reflection is mapped to function extensionality). However there is no *syntactic translation* from extensional type theory to intensional type theory with function extensionality (the [16,26,32] translations do not preserve definitional equalities).

In the following, $|-|_0$ is a model and $|-|_1$, $|-|_2$ and $|-|_3$ are variants which are syntactic translations.

3.1 Standard model

The *standard interpretation* $|-|_0$ (aka set interpretation, or metacircular interpretation) is specified as follows.

$$\frac{\Gamma : \text{Con}_i}{|\Gamma|_0 : \text{Set}_i} \quad \frac{A : \text{Ty}_j \Gamma}{|A|_0 : |\Gamma|_0 \rightarrow \text{Set}_j} \quad \frac{\delta : \text{Sub} \Gamma \Delta}{|\delta|_0 : |\Gamma|_0 \rightarrow |\Delta|_0} \quad \frac{t : \text{Tm} \Gamma A}{|t|_0 : (\gamma : |\Gamma|_0) \rightarrow |A|_0 \gamma}$$

Contexts are mapped to metatheoretic types, types to families of types over the interpretation of the context, substitutions become functions and terms dependent functions. We illustrate the implementation by listing some components for contexts and function space.

$$\begin{aligned} |\cdot|_0 &:= \top \\ |\Gamma, x : A|_0 &:= (\gamma : |\Gamma|_0) \times |A|_0 \gamma \\ |\Pi(x : A). B|_0 \gamma &:= (\alpha : |A|_0 \gamma) \rightarrow |B|_0 (\gamma, \alpha) \\ |\lambda x. t|_0 \gamma \alpha &:= |t|_0 (\gamma, \alpha) \\ |t @ x|_0 (\gamma, \alpha) &:= |t|_0 \gamma \alpha \end{aligned}$$

$$|\Pi\beta|_0 \quad : \quad |(\lambda x.t) @ x|_0 (\gamma, \alpha) = |\lambda x.t|_0 \gamma \alpha = |t|_0 (\gamma, \alpha)$$

The empty context is interpreted by the unit type (note that it is written in black, this is just the metatheoretic unit). Extended contexts are interpreted by metatheoretic Σ types, Π types are interpreted by function space, λ becomes metatheoretic abstraction, $@$ becomes function application, $\Pi\beta$ and $\Pi\eta$ hold by definition.

The standard interpretation $|-|_1$. If we make the metatheory explicit, the previous set interpretation can be seen as a syntactic translation from $\text{MLTT}_{\text{Prop}}$ to $\text{MLTT}_{\text{Prop}}$ extended with a hierarchy of Coquand universes. The latter is no longer called metatheory because the metatheory is now the one in which we talk about both the source and the target theory.

The syntax for Coquand universes⁷ is the following.

$$\frac{}{\mathbf{U}_i : \mathbf{Ty}_{i+1} \Gamma} \quad \frac{A : \mathbf{Ty}_i \Gamma}{\mathbf{c} A : \mathbf{Tm} \Gamma \mathbf{U}_i} \quad \frac{a : \mathbf{Tm} \Gamma \mathbf{U}_i}{\mathbf{El} a : \mathbf{Ty}_i \Gamma} \quad \mathbf{El}(\mathbf{c} A) = A \quad \mathbf{c}(\mathbf{El} a) = a$$

The specification of this interpretation is as follows (we don't distinguish the source and the target theory in our notation).

$$\frac{\Gamma : \mathbf{Con}_i}{|\Gamma|_1 : \mathbf{Tm} \cdot \mathbf{U}_i} \quad \frac{A : \mathbf{Ty}_j \Gamma}{|A|_1 : \mathbf{Tm} \cdot (\mathbf{El} |\Gamma|_1 \Rightarrow \mathbf{U}_j)} \\ \frac{\delta : \mathbf{Sub} \Gamma \Delta}{|\delta|_1 : \mathbf{Tm} \cdot (\mathbf{El} |\Gamma|_1 \Rightarrow \mathbf{El} |\Delta|_1)} \quad \frac{t : \mathbf{Tm} \Gamma A}{|t|_1 : \mathbf{Tm} \cdot (\Pi(\gamma : \mathbf{El} |\Gamma|_1). \mathbf{El} (|A|_1 @ \gamma))}$$

A context becomes a term of type \mathbf{U} in the empty target context. A type becomes a term of a function type with codomain \mathbf{U} . Note the difference between the arrows \rightarrow and \Rightarrow . A substitution becomes a term of function type and a term becomes a term of a dependent function type where we use target theory application $@$ in the codomain.

The difference between $|-|_0$ and $|-|_1$ is that the latter uses an explicit syntax, otherwise the constructions are the same. They both interpret contexts, types, substitutions and terms all as terms. Type dependency is modelled by Π types and comprehension is modelled by Σ types. The interpretation of $\Pi\beta$ illustrates this nicely: apart from the target theory $\Pi\beta$, $\Pi\eta$ is needed for dealing with type dependency and $\Sigma\eta$ for a comprehension law.

$$\begin{aligned} |\cdot|_1 &:= \mathbf{c} \perp \\ |\Gamma, x : A|_1 &:= \mathbf{c} (\Sigma(\gamma : \mathbf{El} |\Gamma|_1). \mathbf{El} (|A|_1 @ \gamma)) \\ |\Pi(x : A). B|_1 &:= \lambda \gamma. \mathbf{c} (\Pi(\alpha : \mathbf{El} (|A|_1 @ \gamma)). \mathbf{El} (|B|_1 @ (\gamma, \alpha))) \\ |\lambda x. t|_1 &:= \lambda \gamma. \lambda \alpha. |t|_1 @ (\gamma, \alpha) \\ |t @ x|_1 &:= \lambda \gamma. |t|_1 @ \mathbf{pr}_0 \gamma @ \mathbf{pr}_1 \gamma \end{aligned}$$

⁷ We learnt this representation of Russell universes from Thierry Coquand.

$$\begin{aligned}
|\Pi\beta|_1 & : \quad |(\lambda x.t) @ x|_1 = \lambda\gamma.|\lambda x.t|_1 @ \text{pr}_0 \gamma @ \text{pr}_1 \gamma = \\
& \quad \lambda\gamma.(\lambda\gamma.\lambda\alpha.|t|_1 @ (\gamma, \alpha)) @ \text{pr}_0 \gamma @ \text{pr}_1 \gamma \stackrel{\Pi\beta}{=} \\
& \quad \lambda\gamma.|t|_1 @ (\text{pr}_0 \gamma, \text{pr}_1 \gamma) \stackrel{\Sigma\eta}{=} \lambda\gamma.|t|_1 @ \gamma \stackrel{\Pi\eta}{=} |t|_1
\end{aligned}$$

$|-|_1$ can be seen as $|-|_0$ composed with a quoting operation returning the syntax of a metatheoretic term (see [31,5]).

Strict vs non-strict models. It is easy to implement the $|-|_0$ interpretation in Agda as it supports all the constructors and equalities of $\text{MLTT}_{\text{Prop}}$, it has Π and Σ types with definitional η laws, and has the required hierarchy of universes. Because all the equalities hold as definitional equalities in Agda, the proofs of these equalities are just reflexivity. We call such a model a *strict model*. A strict model can be always turned into a syntactic translation (to the metatheory as target theory) the same way as we turned $|-|_0$ into $|-|_1$. A non-strict model is one where some of the interpretations of equalities need a proof, that is, they cannot be given by reflexivity.

Note that the notion of strict model is relative to the metatheory. The same model can be strict in one metatheory and not in another one. For example, all models are strict in a metatheory with extensional equality. The standard model $|-|_0$ is strict in Agda, however if we turn off definitional η for Σ types using the pragma `--no-eta`⁸, it becomes non-strict as the definitional η law is needed to interpret the syntactic equality $(\delta, x \mapsto x[\delta]) = \delta$. The model can be still defined because a propositional η law can be proven (the equalities of the model are given by propositional equalities of the metatheory). However this model cannot be turned into a syntactic translation into a target theory with no definitional η for Σ types.

There are models which are not strict, but can be still turned into a syntactic translation. An example is the $_{0a}$ variant of the graph model defined in Subsection 3.2.

We can define two more variants of the standard model by changing what models type dependency and comprehension. $|-|_2$ models type dependency of the source theory with type dependency in the target theory, but still models comprehension using Σ types. $|-|_3$ models type dependency by type dependency and comprehension by comprehension.

The standard interpretation $|-|_2$ does not need a universe in the target theory. In general, this translation works for any source theory once the target theory has \top and Σ types (in addition to all the constructors that the source theory has).

$$\begin{array}{c}
\frac{\Gamma : \text{Con}_i}{|\Gamma|_2 : \text{Ty}_i \cdot} \qquad \frac{A : \text{Ty}_j \Gamma}{|A|_2 : \text{Ty}_j (\cdot, \gamma : |\Gamma|_2)} \\
\\
\frac{\delta : \text{Sub } \Gamma \Delta}{|\delta|_2 : \text{Tm} (\cdot, \gamma : |\Gamma|_2) |\Delta|_2} \qquad \frac{t : \text{Tm } \Gamma A}{|t|_2 : \text{Tm} (\cdot, \gamma : |\Gamma|_2) |A|_2}
\end{array}$$

⁸ Agda version 2.6.0.

The interpretation of $\Pi\beta$ still needs $\Sigma\eta$ because comprehension is given by Σ types, however the type dependency parts are dealt with by substitution laws. For example, we use substitution to write $|B|_2[\gamma \mapsto (\gamma, \alpha)]$ in the interpretation of Π while in the $|-|_1$ variant we used application $|B|_1 @ (\gamma, \alpha)$.

$$\begin{aligned}
|\cdot|_2 &:= \mathbb{I} \\
|\Gamma, x : A|_2 &:= \Sigma(\gamma : |\Gamma|_2). |A|_2 \\
|\Pi(x : A). B|_2 &:= \Pi(\alpha : |A|_2). |B|_2[\gamma \mapsto (\gamma, \alpha)] \\
|\lambda x. t|_2 &:= \lambda \alpha. |t|_2[\gamma \mapsto (\gamma, \alpha)] \\
|t @ x|_2 &:= (|t|_2[\gamma \mapsto \text{pr}_0 \gamma]) @ \text{pr}_1 \gamma \\
|\Pi\beta|_2 &: |(\lambda x. t) @ x|_2 = (|\lambda x. t|_2[\gamma \mapsto \text{pr}_0 \gamma]) @ \text{pr}_1 \gamma = \\
&\quad ((\lambda \alpha. |t|_2[\gamma \mapsto (\gamma, \alpha)])[\gamma \mapsto \text{pr}_0 \gamma]) @ \text{pr}_1 \gamma = \\
&\quad (\lambda \alpha. |t|_2[\gamma \mapsto (\text{pr}_0 \gamma, \alpha)]) @ \text{pr}_1 \gamma \stackrel{\Pi\beta}{=} \\
&\quad |t|_2[\gamma \mapsto (\text{pr}_0 \gamma, \text{pr}_1 \gamma)] \stackrel{\Sigma\eta}{=} |t|_2[\gamma \mapsto \gamma] = |t|_2
\end{aligned}$$

The standard interpretation $|-|_3$ The last variant of the standard interpretation is simply the identity translation: everything is mapped to itself. The source and the target theory can be exactly the same.

$$\frac{\Gamma : \text{Con}_i}{|\Gamma|_3 : \text{Con}_i} \quad \frac{A : \text{Ty}_j \Gamma}{|A|_3 : \text{Ty}_j |\Gamma|_3} \quad \frac{\delta : \text{Sub } \Gamma \Delta}{|\delta|_3 : \text{Sub } |\Gamma|_3 |\Delta|_3} \quad \frac{t : \text{Tm } \Gamma A}{|t|_3 : \text{Tm } |\Gamma|_3 |A|_3}$$

Here the interpretation of $\Pi\beta$ obviously only needs $\Pi\beta$.

$$\begin{aligned}
|\cdot|_3 &:= \cdot \\
|\Gamma, x : A|_3 &:= |\Gamma|_3, x : |A|_3 \\
|\Pi(x : A). B|_3 &:= \Pi(x : |A|_3). |B|_3 \\
|\lambda x. t|_3 &:= \lambda x. |t|_3 \\
|t @ x|_3 &:= |t|_3 @ x \\
|\Pi\beta|_3 &: |(\lambda x. t) @ x|_3 = (\lambda x. |t|_3) @ x \stackrel{\Pi\beta}{=} |t|_3
\end{aligned}$$

3.2 Graph model

In this subsection we define variants of the graph model corresponding to the $_0$, $_1$, $_2$, $_3$ variants of the standard model. Here each syntactic component is mapped to two components $|-|$ and $-\sim$. The $|-|$ components are the same as in the case of the standard model.

The metatheoretic $_0$ variant of the graph interpretation is specified as follows.

$$\begin{array}{c}
\frac{\Gamma : \text{Con}_i}{| \Gamma |_0 : \text{Set}_i} \\
\frac{}{\Gamma^{\sim 0} : | \Gamma |_0 \rightarrow | \Gamma |_0 \rightarrow \text{Set}_i}
\end{array}
\qquad
\begin{array}{c}
\frac{A : \text{Ty}_j \Gamma}{| A |_0 : | \Gamma |_0 \rightarrow \text{Set}_j} \\
\frac{}{A^{\sim 0} : \Gamma^{\sim 0} \gamma_0 \gamma_1 \rightarrow} \\
\frac{}{| A |_0 \gamma_0 \rightarrow | A |_0 \gamma_1 \rightarrow \text{Set}_j}
\end{array}$$

$$\begin{array}{c}
\frac{\delta : \text{Sub } \Gamma \Delta}{| \delta |_0 : | \Gamma |_0 \rightarrow | \Delta |_0} \\
\frac{}{\delta^{\sim 0} : \Gamma^{\sim 0} \gamma_0 \gamma_1 \rightarrow \Delta^{\sim 0} (| \delta |_0 \gamma_0) (| \delta |_0 \gamma_1)}
\end{array}
\qquad
\begin{array}{c}
\frac{t : \text{Tm } \Gamma A}{| t |_0 : (\gamma : | \Gamma |_0) \rightarrow | A |_0 \gamma} \\
\frac{}{t^{\sim 0} : (\gamma_{01} : \Gamma^{\sim 0} \gamma_0 \gamma_1) \rightarrow} \\
\frac{}{A^{\sim 0} \gamma_{01} (| t |_0 \gamma_0) (| t |_0 \gamma_1)}
\end{array}$$

Models of type theory are usually named after what contexts are mapped to (a set for the set model, a setoid for the setoid model, etc.). In the graph model a context is mapped to a graph: a set of vertices and for every two vertex, a set of arrows between those. In short, a set and a proof-relevant binary relation over it. Types are interpreted as displayed graphs over a base graph: a family over each vertex and a heterogeneous binary relation over each arrow. Substitutions become graph homomorphisms and terms their displayed variants. Note that in the types of $A^{\sim 0}$, $\delta^{\sim 0}$ and $t^{\sim 0}$, we implicitly quantified over γ_0 and γ_1 .

Variant $_1$ of the graph interpretation is specified as follows. Again, we need a Coquand universe \mathbf{U} in the target.

$$\begin{array}{c}
\frac{\Gamma : \text{Con}_i}{| \Gamma |_1 : \text{Tm} \cdot \mathbf{U}_i} \\
\frac{}{\Gamma^{\sim 1} : \text{Tm} \cdot (\text{El } | \Gamma |_1 \Rightarrow \text{El } | \Gamma |_1 \Rightarrow \mathbf{U}_i)}
\end{array}$$

$$\begin{array}{c}
\frac{A : \text{Ty}_i \Gamma}{| A |_1 : \text{Tm} \cdot (\text{El } | \Gamma |_1 \Rightarrow \text{El } | \Delta |_1)} \\
\frac{}{A^{\sim 1} : \text{Tm} \cdot (\text{El } (\Gamma^{\sim 1} @ \gamma_0 @ \gamma_1) \Rightarrow \text{El } (| A |_1 @ \gamma_0) \Rightarrow \text{El } (| A |_1 @ \gamma_1) \Rightarrow \mathbf{U}_i)}
\end{array}$$

$$\begin{array}{c}
\frac{\delta : \text{Sub } \Gamma \Delta}{| \delta |_1 : \text{Tm} \cdot (\text{El } | \Gamma |_1 \Rightarrow \mathbf{U}_i)} \\
\frac{}{\delta^{\sim 1} : \text{Tm} \cdot (\text{El } (\Gamma^{\sim 1} @ \gamma_0 @ \gamma_1) \Rightarrow \text{El } (\Delta^{\sim 1} @ (| \delta |_1 @ \gamma_0) @ (| \delta |_1 @ \gamma_1)))}
\end{array}$$

$$\begin{array}{c}
\frac{t : \text{Tm } \Gamma A}{| t |_1 : \text{Tm} \cdot (\text{El } (\gamma : \text{El } | \Gamma |_1) . \text{El } (| A |_1 @ \gamma))} \\
\frac{}{t^{\sim 1} : \text{Tm} \cdot (\text{El } (\gamma_{01} : \text{El } (\Gamma^{\sim 1} @ \gamma_0 @ \gamma_1)) . \text{El } (A^{\sim 1} @ \gamma_{01} @ (| t |_1 @ \gamma_0) @ (| t |_1 @ \gamma_1)))}
\end{array}$$

The relations become target theory terms which have function types with codomain \mathbf{U} . We used implicit quantification in the target theory for ease of reading. For example, the type of $\delta^{\sim 1}$ should be understood as

$$\text{Tm} \cdot (\text{El } (\gamma_0 : | \Gamma |_1) . \text{El } (\gamma_0 : | \Gamma |_1) . \text{El } (\Gamma^{\sim 1} @ \gamma_0 @ \gamma_1) \Rightarrow \Delta^{\sim 1} @ (| \delta |_1 @ \gamma_0) @ (| \delta |_1 @ \gamma_1)).$$

Variant $_2$ of the graph interpretation is specified as follows.

$$\begin{array}{c}
\frac{\Gamma : \mathbf{Con}_i}{| \Gamma |_2 : \mathbf{Ty}_i \cdot} \\
\frac{}{\Gamma^{\sim 2} : \mathbf{Ty}_i(\cdot, \gamma_0 : | \Gamma |_2, \gamma_1 : | \Gamma |_2)} \\
\\
\frac{A : \mathbf{Ty}_i \Gamma}{| A |_2 : \mathbf{Ty}_i(\cdot, \gamma : | \Gamma |_2)} \\
\frac{}{A^{\sim 2} : \mathbf{Ty}_i(\cdot, \gamma_{01} : \Gamma^{\sim 2}, \alpha_0 : | A |_2[\gamma \mapsto \gamma_0], \alpha_1 : | A |_2[\gamma \mapsto \gamma_1])} \\
\\
\frac{\delta : \mathbf{Sub} \Gamma \Delta}{| \delta |_2 : \mathbf{Tm}(\cdot, \gamma : | \Gamma |_2) | \Delta |_2} \\
\frac{}{\delta^{\sim 2} : \mathbf{Tm}(\cdot, \gamma_{01} : \Gamma^{\sim 2}) (\Delta^{\sim 2} [\gamma_0 \mapsto | \delta |_2[\gamma \mapsto \gamma_0], \gamma_1 \mapsto | \delta |_2[\gamma \mapsto \gamma_1]])} \\
\\
\frac{t : \mathbf{Tm} \Gamma A}{| t |_2 : \mathbf{Tm}(\cdot, \gamma : | \Gamma |_2) | A |_2} \\
\frac{}{t^{\sim 2} : \mathbf{Tm}(\cdot, \gamma_{01} : \Gamma^{\sim 2}) (A^{\sim 2} [\alpha_0 \mapsto | t |_2[\gamma \mapsto \gamma_0], \alpha_1 \mapsto | t |_2[\gamma \mapsto \gamma_1]])}
\end{array}$$

This variant shows that the extra \sim^2 components in the model can be expressed without using Π or a universe: type dependency is enough to express e.g. that $\Gamma^{\sim 2}$ is indexed over two copies of $| \Gamma |_2$. Here, analogously to the usage of implicit Π s in variant $_1$, we use implicit context extensions in the target theory. For example, the type of $\delta^{\sim 2}$ should be understood as

$$\mathbf{Tm}(\cdot, \gamma_0 : | \Gamma |_2, \gamma_1 : | \Gamma |_2, \gamma_{01} : \Gamma^{\sim 2}) (\Delta^{\sim 2} [\gamma_0 \mapsto | \delta |_2[\gamma \mapsto \gamma_0], \gamma_1 \mapsto | \delta |_2[\gamma \mapsto \gamma_1]]).$$

Defining variant $_3$ of the graph interpretation is not as straightforward as the previous ones. As $| \Gamma |_3 : \mathbf{Con}$, we need a notion of binary relation over a context. One solution is going back to the $| - |_0$ model and using the equivalence between indexed families and fibrations [11, p. 221]:

$$A \rightarrow \mathbf{Set} \simeq (A' : \mathbf{Set}) \times (A' \rightarrow A).$$

This means that we replace $\Gamma^{\sim 0} : | \Gamma |_0 \rightarrow | \Gamma |_0 \rightarrow \mathbf{Set}$ with a set $\Gamma^{\sim 0a}$ and two projections $0_{0a}, 1_{0a}$ which give the domain and codomain of the arrow. This interpretation is specified as follows (the $| - |$ components are the same as in the $_0$ model, so they don't have the $_a$ subscript).

$$\begin{array}{c}
\frac{\Gamma : \mathbf{Con}_i}{| \Gamma |_0 : \mathbf{Set}_i} \\
\frac{}{\Gamma^{\sim 0a} : \mathbf{Set}_i} \\
\frac{}{0_{0a} \Gamma : \Gamma^{\sim 0a} \rightarrow | \Gamma |_0} \\
\frac{}{1_{0a} \Gamma : \Gamma^{\sim 0a} \rightarrow | \Gamma |_0} \\
\\
\frac{A : \mathbf{Ty}_i \Gamma}{| A |_0 : | \Gamma |_0 \rightarrow \mathbf{Set}_i} \\
\frac{}{A^{\sim 0a} : (\gamma_{01} : \Gamma^{\sim 0a}) \rightarrow | A |_0 (0_{0a} \Gamma \gamma_{01}) \rightarrow | A |_0 (1_{0a} \Gamma \gamma_{01}) \rightarrow \mathbf{Set}_i}
\end{array}$$

$$\begin{array}{c}
\delta : \text{Sub } \Gamma \Delta \\
\hline
|\delta|_0 : |\Gamma|_0 \rightarrow |\Delta|_0 \\
\delta^{\sim_{0a}} : \Gamma^{\sim_{0a}} \rightarrow \Delta^{\sim_{0a}} \\
0_{0a} \delta : (\gamma_{01} : \Gamma^{\sim_{0a}}) \rightarrow 0_{0a} \Delta (\delta^{\sim_{0a}} \gamma_{01}) = |\delta|_0 (0_{0a} \Gamma \gamma_{01}) \\
1_{0a} \delta : (\gamma_{01} : \Gamma^{\sim_{0a}}) \rightarrow 1_{0a} \Delta (\delta^{\sim_{0a}} \gamma_{01}) = |\delta|_0 (1_{0a} \Gamma \gamma_{01}) \\
\hline
t : \text{Tm } \Gamma A \\
\hline
|t|_0 : (\gamma : |\Gamma|_0) \rightarrow |A|_0 \gamma \\
t^{\sim_{0a}} : (\gamma_{01} : \Gamma^{\sim_{0a}}) \rightarrow A^{\sim_{0a}} \gamma_{01} (|t|_0 (0_{0a} \Gamma \gamma_{01})) (|t|_0 (1_{0a} \Gamma \gamma_{01}))
\end{array}$$

The fact that contexts are given as fibrations forces substitutions to include some equalities, while types are still indexed. This is an example of a model which can be turned into a syntactic translation, but is not strict in Agda. The reason is that equalities are needed to interpret substitutions and in turn we use these equalities to interpret some conversion rules. For example, the 0_{0a} and 1_{0a} components in the interpretation of $- \circ -$ are given by transitivity of equality, so associativity of substitutions needs associativity of transitivity (or UIP). We believe however that this model would be strict in a setoid type theory (Section 5).

In the corresponding $_{3a}$ variant a context is interpreted by two contexts and two projection substitutions. This is the same as the parametricity translation of Bernardy et al [7]. We list the $|-|_3$ part separately because we will reuse it in \sim_{3b} .

$$\begin{array}{c}
\frac{\Gamma : \text{Con}_i}{|\Gamma|_3 : \text{Con}_i} \quad \frac{A : \text{Ty}_i \Gamma}{|A|_3 : \text{Ty}_i |\Gamma|_3} \quad \frac{\delta : \text{Sub } \Gamma \Delta}{|\delta|_3 : \text{Sub } |\Gamma|_3 |\Delta|_3} \quad \frac{t : \text{Tm } \Gamma A}{|t|_3 : \text{Tm } |\Gamma|_3 |A|_3} \\
\hline
\begin{array}{cc}
\frac{\Gamma : \text{Con}_i}{\Gamma^{\sim_{3a}} : \text{Con}_i} & \frac{\delta : \text{Sub } \Gamma \Delta}{\delta^{\sim_{3a}} : \text{Sub } \Gamma^{\sim_{3a}} \Delta^{\sim_{3a}}} \\
0_{3a} \Gamma : \text{Sub } \Gamma^{\sim_{3a}} |\Gamma|_3 & 0_{3a} \delta : 0_{3a} \Delta \circ \delta^{\sim_{3a}} = |\delta|_3 \circ 0_{3a} \Gamma \\
1_{3a} \Gamma : \text{Sub } \Gamma^{\sim_{3a}} |\Gamma|_3 & 1_{3a} \delta : 1_{3a} \Delta \circ \delta^{\sim_{3a}} = |\delta|_3 \circ 1_{3a} \Gamma
\end{array} \\
\hline
\frac{A : \text{Ty}_i \Gamma}{A^{\sim_{3a}} : \text{Ty}_i (\Gamma^{\sim_{3a}}, \alpha_0 : |A|_3 [0_{3a} \Gamma], \alpha_1 : |A|_3 [1_{3a} \Gamma])} \\
\hline
\frac{t : \text{Tm } \Gamma A}{t^{\sim_{3a}} : \text{Tm } \Gamma^{\sim_{3a}} (A^{\sim_{3a}} [\alpha_0 \mapsto |t|_3 [0_{3a} \Gamma], \alpha_1 \mapsto |t|_3 [1_{3a} \Gamma]])}
\end{array}$$

The $_{3b}$ variant of the graph interpretation. Another solution is to define Γ^{\sim_3} in an indexed way by referring to substitutions into $|\Gamma|_3$. This is how we define \sim_{3b} . The $|-|_3$ parts are the same as in $_{3a}$.

$$\begin{array}{c}
\frac{\Gamma : \text{Con}_i \quad \rho_0, \rho_1 : \text{Sub } \Omega |\Gamma|_3}{\Gamma^{\sim_{3b}} \rho_0 \rho_1 : \text{Ty}_i \Omega} \\
(\Gamma^{\sim_{3b}} \rho_0 \rho_1)[\nu] = \Gamma^{\sim_{3b}} (\rho_0 \circ \nu) (\rho_1 \circ \nu) \\
\hline
\frac{A : \text{Ty}_i \Gamma \quad \rho_{01} : \Gamma^{\sim_{3b}} \rho_0 \rho_1 \quad t_0 : \text{Tm } \Omega (|A|_3 [\rho_0]) \quad t_1 : \text{Tm } \Omega (|A|_3 [\rho_1])}{A^{\sim_{3b}} \rho_{01} t_0 t_1 : \text{Ty}_i \Omega} \\
(A^{\sim_{3b}} \rho_{01} t_0 t_1)[\nu] = A^{\sim_{3b}} (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu])
\end{array}$$

$$\begin{array}{c}
\frac{\delta : \text{Sub } \Gamma \Delta \quad \rho_{01} : \text{Tm } \Omega (\Gamma^{\sim_{3b}} \rho_0 \rho_1)}{\delta^{\sim_{3b}} \rho_{01} : \text{Tm } \Omega (\Delta^{\sim_{3b}} (|\delta|_3 \circ \rho_0) (|\delta|_3 \circ \rho_1))} \\
(\delta^{\sim_{3b}} \rho_{01})[\nu] = \delta^{\sim_{3b}} (\rho_{01}[\nu]) \\
\\
\frac{t : \text{Tm } \Gamma A \quad \rho_{01} : \text{Tm } \Omega (\Gamma^{\sim_{3b}} \rho_0 \rho_1)}{t^{\sim_{3b}} \rho_{01} : \text{Tm } \Omega (A^{\sim_{3b}} \rho_{01} (|t|_3 [\rho_0]) (|t|_3 [\rho_1]))} \\
(t^{\sim_{3b}} \rho_{01})[\nu] = t^{\sim_{3b}} (\rho_{01}[\nu])
\end{array}$$

Ω , ρ_0 and ρ_1 are implicit parameters of $A^{\sim_{3b}}$, $\delta^{\sim_{3b}}$ and $t^{\sim_{3b}}$. The advantage of the \sim_{3b} compared to the \sim_{3a} is that we don't need the projection substitutions for contexts and the naturality conditions for substitutions, the disadvantage is that we need the extra equalities expressing substitution laws.

4 The setoid model as a translation

In this section, after recalling the setoid model, we turn it into a syntactic translation from $\text{MLTT}_{\text{Prop}}$ to $\text{MLTT}_{\text{Prop}}$. We follow the approach of graph model variant $_{3b}$ (Section 3.2) and extend it into a setoid syntactic translation where a context is modelled not only by a set and a relation, but a set and an equivalence relation.

4.1 The setoid model

In the setoid model [1], a context is given by a set together with an equivalence relation which, in contrast with the graph model, is proof-irrelevant. We think about this relation as the explicit equality relation for the set. A type is interpreted by a displayed setoid together with a coercion and coherence operation. Coercion transports between families at related objects and coherence says that this coercion respects the displayed relation.

$\Gamma : \text{Con}_i$	$A : \text{Ty}_j \Gamma$
$ \Gamma _0 : \text{Set}_i$	$ A _0 : \Gamma _0 \rightarrow \text{Set}_j$
$\Gamma^{\sim_0} : \Gamma _0 \rightarrow \Gamma _0 \rightarrow \text{Prop}_i$	$A^{\sim_0} : \Gamma^{\sim_0} \gamma_0 \gamma_1 \rightarrow A _0 \gamma_0 \rightarrow A _0 \gamma_1 \rightarrow \text{Prop}_j$
$\text{R}_\Gamma^0 : (\gamma : \Gamma _0) \rightarrow \Gamma^{\sim_0} \gamma \gamma$	$\text{R}_A^0 : (\alpha : A _0 \gamma) \rightarrow A^{\sim_0} (\text{R}_\Gamma^0 \gamma) \alpha \alpha$
$\text{S}_\Gamma^0 : \Gamma^{\sim_0} \gamma_0 \gamma_1 \rightarrow \Gamma^{\sim_0} \gamma_1 \gamma_0$	$\text{S}_A^0 : A^{\sim_0} \gamma_{01} \alpha_0 \alpha_1 \rightarrow A^{\sim_0} (\text{S}_\Gamma^0 \gamma_{01}) \alpha_1 \alpha_0$
$\text{T}_\Gamma^0 : \Gamma^{\sim_0} \gamma_0 \gamma_1 \rightarrow$ $\Gamma^{\sim_0} \gamma_1 \gamma_2 \rightarrow$ $\Gamma^{\sim_0} \gamma_0 \gamma_2$	$\text{T}_A^0 : A^{\sim_0} \gamma_{01} \alpha_0 \alpha_1 \rightarrow A^{\sim_0} \gamma_{12} \alpha_1 \alpha_2 \rightarrow$ $A^{\sim_0} (\text{T}_\Gamma^0 \gamma_{01} \gamma_{12}) \alpha_0 \alpha_2$
	$\text{coe}_A^0 : \Gamma^{\sim_0} \gamma_0 \gamma_1 \rightarrow A _0 \gamma_0 \rightarrow A _0 \gamma_1$
	$\text{coh}_A^0 : (\gamma_{01} : \Gamma^{\sim_0} \gamma_0 \gamma_1) \rightarrow (\alpha_0 : A _0 \gamma_0) \rightarrow$ $A^{\sim_0} \gamma_{01} \alpha_0 (\text{coe}_A^0 \gamma_{01} \alpha_0)$

This notion of family of setoids is different from Altenkirch's original one [1] but is equivalent to it [9, Section 1.6.1]. Substitutions and terms are specified the same as in the graph model (see $|-|_0$ in Section 3.2). There is no need for **R**, **S**, **T** components because these are provable by proof irrelevance (unlike in the groupoid model [19,24]).

4.2 Specification of the translation

In the following we turn the setoid interpretation $_0$ into a setoid syntactic translation following the $_{3b}$ variant of the graph translation. We drop the $_{3b}$ indices to ease the reading. We expect the other variants to be definable as well.

An $\text{MLTT}_{\text{Prop}}$ context is mapped to six components: a context, a binary propositional relation over substitutions into that context, reflexivity, symmetry and transitivity of this relation and a substitution law for \sim . Note that we use implicit arguments, e.g. $\Gamma \sim$ takes an $\Omega : \text{Con}$ implicitly and S_Γ takes $\Omega : \text{Con}$, $\rho_0, \rho_1 : \text{Sub } \Omega \mid \Gamma \mid$ implicitly.

$$\begin{array}{c}
 \Gamma : \text{Con}_i \\
 \hline
 \mid \Gamma \mid : \text{Con}_i \\
 \Gamma \sim : \text{Sub } \Omega \mid \Gamma \mid \rightarrow \text{Sub } \Omega \mid \Gamma \mid \rightarrow \text{Tm } \Omega \text{ Prop}_i \\
 \Gamma \sim [] : (\Gamma \sim \rho_0 \rho_1)[\nu] = \Gamma \sim (\rho_0 \circ \nu) (\rho_1 \circ \nu) \\
 R_\Gamma : (\rho : \text{Sub } \Omega \mid \Gamma \mid) \rightarrow \text{Tm } \Omega \Gamma \sim \rho \rho \\
 S_\Gamma : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1 \rightarrow \text{Tm } \Omega \Gamma \sim \rho_1 \rho_0 \\
 T_\Gamma : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1 \rightarrow \text{Tm } \Omega \Gamma \sim \rho_1 \rho_2 \rightarrow \text{Tm } \Omega \Gamma \sim \rho_0 \rho_2
 \end{array}$$

A type is interpreted by a type over the interpretation of the context, a heterogeneous relation over the relation for contexts which is reflexive, symmetric and transitive (over the corresponding proofs for the contexts). Moreover, there is a coercion function which relates types substituted by related substitutions. Coherence expresses that coercion respects the relation (coh). The \sim relation and coe come with substitution laws.

$$\begin{array}{c}
 A : \text{Ty}_j \Gamma \\
 \hline
 \mid A \mid : \text{Ty}_j \mid \Gamma \mid \\
 A \sim : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1 \rightarrow \text{Tm } \Omega (\mid A \mid [\rho_0]) \rightarrow \text{Tm } \Omega (\mid A \mid [\rho_1]) \rightarrow \text{Tm } \Omega \text{ Prop}_j \\
 A \sim [] : (A \sim \rho_{01} t_0 t_1)[\nu] = A \sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) \\
 R_A : (t : \text{Tm } \Omega (\mid A \mid [\rho])) \rightarrow \text{Tm } \Omega A \sim (R_\Gamma \rho) t t \\
 S_A : \text{Tm } \Omega A \sim \rho_{01} t_0 t_1 \rightarrow \text{Tm } \Omega A \sim (S_\Gamma \rho_{01}) t_1 t_0 \\
 T_A : \text{Tm } \Omega A \sim \rho_{01} t_0 t_1 \rightarrow \text{Tm } \Omega A \sim \rho_{12} t_1 t_2 \rightarrow \text{Tm } \Omega A \sim (T_\Gamma \rho_{01} \rho_{12}) t_0 t_2 \\
 \text{coe}_A : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1 \rightarrow \text{Tm } \Omega (\mid A \mid [\rho_0]) \rightarrow \text{Tm } \Omega (\mid A \mid [\rho_1]) \\
 \text{coe}[]_A : (\text{coe}_A \rho_{01} t_0)[\nu] = \text{coe}_A (\rho_{01}[\nu]) (t_0[\nu]) \\
 \text{coh}_A : (\rho_{01} : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1) (t_0 : \text{Tm } \Omega (\mid A \mid [\rho_0])) \rightarrow \\
 \text{Tm } \Omega A \sim \rho_{01} t_0 (\text{coe}_A \rho_{01} t_0)
 \end{array}$$

A substitution is interpreted by a substitution which respects the relations.

$$\begin{array}{c}
 \delta : \text{Sub } \Gamma \Delta \\
 \hline
 \mid \delta \mid : \text{Sub } \mid \Gamma \mid \mid \Delta \mid \\
 \delta \sim : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1 \rightarrow \text{Tm } \Omega \Delta \sim (\mid \delta \mid \circ \rho_0) (\mid \delta \mid \circ \rho_1)
 \end{array}$$

A term is interpreted by a term which respects the relations.

$$\begin{array}{c}
 t : \text{Tm } \Gamma A \\
 \hline
 \mid t \mid : \text{Tm } \mid \Gamma \mid \mid A \mid \\
 t \sim : (\rho_{01} : \text{Tm } \Omega \Gamma \sim \rho_0 \rho_1) \rightarrow \text{Tm } \Omega A \sim \rho_{01} (\mid t \mid [\rho_0]) (\mid t \mid [\rho_1])
 \end{array}$$

Note that we do not need substitution laws for those components which don't have any parameters (the $|-|$ ones) and those which result in a term of an underlined type. The laws for the latter ones hold by proof irrelevance.

4.3 Implementation of the translation

We implement this specification by explaining what the different components of $\text{MLTT}_{\text{Prop}}$ are mapped to. All details can be found in Appendix B. As in the case of the $|-|_3$ standard and graph interpretations, the $|-|$ components in the setoid translation are almost always identity functions. The only exception is the case of Π where a function is interpreted by a function which preserves equality:

$$\begin{aligned} |\Pi(x : A).B| &:= \Sigma(f : \Pi(x : |A|).|B|). \underline{\pi(x_0 : |A|). \pi(x_1 : |A|)}. \\ &\quad \underline{\pi(x_{01} : A \sim (\text{R}_\Gamma \text{id}) x_0 x_1). B \sim (\text{R}_\Gamma \text{id}, x_{01}) (f @ x_0) (f @ x_1)} \end{aligned}$$

Equality of functions is defined by saying that the first component (pr_0) of the interpretation of the function preserves equality:

$$\begin{aligned} (\Pi(x : A).B) \sim \rho_{01} t_0 t_1 &:= \pi(x_0 : |A|[\rho_0]). \pi(x_1 : |A|[\rho_1]). \pi(x_{01} : A \sim \rho_{01} x_0 x_1). \\ &\quad B \sim (\rho_{01}, x_{01}) (\text{pr}_0 t_0 @ x_0) (\text{pr}_0 t_1 @ x_1), \end{aligned}$$

We need the second component to implement reflexivity for the function space: $\text{R}_{\Pi(x : A).B} t := \text{pr}_1 t$. Equality for extended contexts and Σ types is pointwise, equality of booleans is given by a decision procedure, equality of propositions is logical equivalence and equality of proofs of propositions is trivial:

$$\begin{aligned} (\Gamma, x : A) \sim (\rho_0, x \mapsto t_0) (\rho_1, x \mapsto t_1) &:= \sigma(\rho_{01} : \Gamma \sim \rho_0 \rho_1). A \sim \rho_{01} t_0 t_1 \\ (\Sigma(x : A).B) \sim \rho_{01} (u_0, v_0) (u_1, v_1) &:= \sigma(x_{01} : A \sim \rho_{01} u_0 u_1). B \sim (\rho_{01}, x_{01}) v_0 v_1 \\ \text{Bool} \sim \rho_{01} t_0 t_1 &:= \text{if } t_0 \text{ then (if } t_1 \text{ then } \top \text{ else } \perp) \\ &\quad \text{else (if } t_1 \text{ then } \perp \text{ else } \top) \\ \text{Prop}_i \sim \rho_{01} a_0 a_1 &:= (\underline{a_0} \Rightarrow a_1) \times (\underline{a_1} \Rightarrow a_0) \\ \underline{a} \sim \rho_{01} t_0 t_1 &:= \top \end{aligned}$$

Symmetry for Π types takes as input an equality proof x_{01} , applies symmetry on it at the domain type, then applies the proof of equality, then applies symmetry at the codomain type: $\text{S}_{\Pi(x : A).B} t_{01} := \lambda x_0 x_1 x_{01}. \text{S}_B (t_{01} @ x_1 @ x_0 @ \text{S}_A x_{01})$. Coercion needs to produce a function $t_0 : \text{tm } \Omega(|\Pi(x : A).B|[\rho_0])$ and has to produce one of type $|\Pi(x : A).B|[\rho_1]$. The first component is given by

$$\lambda x_1. \text{coe}_B (\rho_{01}, \text{coh}_A (\text{S}_\Gamma \rho_{01}) x_1) (\text{pr}_0 t_0 @ \text{coe}_A (\text{S}_\Gamma \rho_{01}) x_1).$$

First the input is coerced backwards by coe_A^* (from $|A|[\rho_1]$ to $|A|[\rho_0]$), then the function is applied, then the output is coerced forwards by coe_B . Backwards coercion coe_A^* is defined using coe_A and S_Γ . Backwards coherence is defined in a similar way, see Appendix B.2 for details.

Reflexivity, symmetry and transitivity are pointwise for Σ types. For **Bool**, they are defined using **if-then-else**-, e.g. $R_{\text{Bool}} t := \text{if } t \text{ then tt else tt}$. As **Bool** is a closed type, coercion is the identity function and coherence is trivial.

Reflexivity for propositions is given by two identity functions: $R_{\text{Prop}_i} a := (\lambda x.x, \lambda x.x)$. Symmetry swaps the functions: $S_{\text{Prop}_i} (a_{01}, a_{10}) := (a_{10}, a_{01})$. Coercion is the identity, and hence coherence is given by two identity functions. For \underline{a} types, reflexivity, symmetry, transitivity and coherence are all trivial (**tt**). Coercion is \underline{a} is more interesting: it uses a function from the logical equivalence given by $a \sim \rho_{01}$.

$$\text{coe}_{\underline{a}} \rho_{01} t_0 := \text{pr}_0 (a \sim \rho_{01}) @ t_0$$

The rest of the setoid translation follows that of the setoid model [1], see Appendix B for all the details.

4.4 Extensions

The identity type. We extend the signature of $\text{MLTT}_{\text{Prop}}$ given in Section 2 with Martin-Löf's inductive identity type with a propositional computation rule, function extensionality and propositional extensionality by the following rules.

$$\begin{array}{c} \frac{A : \text{Ty}_i \Gamma \quad u, v : \text{Tm } \Gamma A}{\text{Id}_A u v : \text{Tm } \Gamma \text{Prop}_i} \\[10pt] \frac{P : \text{Ty}_i (\Gamma, x : A) \quad e : \text{Tm } \Gamma \text{Id}_A u v \quad t : \text{Tm } \Gamma (P[x \mapsto u])}{\text{transport}_{x.P} e t : \text{Tm } \Gamma (P[x \mapsto v])} \\[10pt] \frac{u : \text{Tm } \Gamma A}{\text{refl}_u : \text{Tm } \Gamma \text{Id}_A u u} \quad \frac{P : \text{Ty}_i (\Gamma, x : A) \quad t : \text{Tm } \Gamma (P[x \mapsto u])}{\text{Id}_\beta t : \text{Tm } \Gamma \text{Id}_{P[x \mapsto u]} (\text{transport}_{x.P} \text{refl}_u t) t} \\[10pt] \frac{t_0, t_1 : \text{Tm } \Gamma (\Pi(x : A).B) \quad e : \text{Tm } \Gamma (\Pi(x : A). \text{Id}_B (t_0 @ x) (t_1 @ x))}{\text{funext } e : \text{Tm } \Gamma \text{Id}_{\Pi(x : A).B} t_0 t_1} \\[10pt] \frac{a_0, a_1 : \text{Tm } \Gamma \text{Prop} \quad t : \text{Tm } \Gamma (\underline{a_0} \Rightarrow a_1) \times (\underline{a_1} \Rightarrow a_0)}{\text{propext } t : \text{Tm } \Gamma \text{Id}_{\text{Prop}} a_0 a_1} \end{array}$$

$$\begin{array}{lcl} \text{Id}[] & : (\text{Id}_A u v)[\nu] & = \text{Id}_{A[\nu]} (u[\nu]) (v[\nu]) \\ \text{transport}[] & : (\text{transport}_{x.P} e t)[\nu] & = \text{transport}_{x.P[\nu]} (e[\nu]) (t[\nu]) \end{array}$$

Note that the dependent eliminator for **Id** (usually called **J**) can be derived from **transport** in the presence of UIP (as in our setting).

The setoid translation given in Subsections 4.2–4.3 translates from $\text{MLTT}_{\text{Prop}}$ to $\text{MLTT}_{\text{Prop}}$. However it extends to a translation from $\text{MLTT}_{\text{Prop}}$ + identity type to $\text{MLTT}_{\text{Prop}}$. $|\text{Id}_A u v|$ is defined as $A \sim (R_\Gamma \text{id}) |u| |v|$, $|\text{transport}_{x.P} e t|$ is given by $\text{coe}_P (R_\Gamma \text{id}, |e|) |t|$. Function extensionality and propositional extensionality are also justified. See Appendix B.3 for the translation of all the rules of the identity type.

Definitional computation rule for transport. We can extend the setoid translation with the following new component for types:

$$\frac{A : \mathbf{Ty}_i \Gamma}{\text{coe}_{R_A} : (\rho : \mathbf{Sub} \Omega | \Gamma |)(t : \mathbf{Tm} \Omega (|A|[\rho])) \rightarrow \text{coe}_A (R_\Gamma \rho) t = t}$$

This expresses that coercion along reflexivity is the identity. Once we have this, the propositional computation rule of **transport** becomes definitional:

$$\begin{aligned} |\mathbf{transport}_{x.P} \text{refl}_u t| &= \\ \text{coe}_P (R_\Gamma \text{id}_{| \Gamma |}, |\text{refl}_u|) |t| &\stackrel{\text{irr}}{=} \\ \text{coe}_P (R_{\Gamma, x:A} \text{id}_{| \Gamma, x:A |}) |t| &\stackrel{\text{coe}_{R_P} \text{id}}{=} |t| \end{aligned}$$

Adding this rule to the setoid translation amounts to checking whether this equality holds for all type formers, we do this in Appendix B.3. Our Agda formalisation of the setoid model [21] also includes this rule and no axioms are required to justify it.

5 Setoid type theory

In this section we extend the signature of $\text{MLTT}_{\text{Prop}}$ given in Section 2 with a new heterogeneous equality type. This extended algebraic theory is called *setoid type theory*. The heterogeneous equality type is inspired by the setoid translation of the previous section. The idea is that we simply add the rules of the setoid translation as new term formers to $\text{MLTT}_{\text{Prop}}$. If we did this naively, this would mean adding the operations $|-|$, \sim , R , S , T , coe , coh as new syntax and all the $:=$ definitions of Section 4.3 as definitional equalities to the syntax. However this would not result in a usable type theory: $A \sim$ would not be a relation between terms of type A , but terms of type $|A|$, so we wouldn't even have a general identity type. Our solution is to *not* add $|-|$ as new syntax (as it is mostly the identity anyway), but only the other components. Moreover, we only add those equalities from the translation which are not derivable by **irr**.

Thus we extend $\text{MLTT}_{\text{Prop}}$ with the following new constructors which explain equality of contexts. This is a homogeneous equivalence relation on substitutions into the context. These new constructors follow the components Γ^\sim , R_Γ , S_Γ , T_Γ in the setoid translation (Section 4.2) except that they do not refer to $|-|$.

$$\begin{aligned} &\frac{\Gamma : \mathbf{Con}_i \quad \rho_0, \rho_1 : \mathbf{Sub} \Omega \Gamma}{\Gamma^\sim \rho_0 \rho_1 : \mathbf{Tm} \Omega \mathbf{Prop}_i} \quad \frac{\Gamma : \mathbf{Con}_i \quad \rho : \mathbf{Sub} \Omega \Gamma}{R_\Gamma \rho : \mathbf{Tm} \Omega \Gamma^\sim \rho \rho} \\ &\frac{\Gamma : \mathbf{Con}_i \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma^\sim \rho_0 \rho_1}{S_\Gamma \rho_{01} : \mathbf{Tm} \Omega \Gamma^\sim \rho_1 \rho_0} \\ &\frac{\Gamma : \mathbf{Con}_i \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma^\sim \rho_0 \rho_1 \quad \rho_{12} : \mathbf{Tm} \Omega \Gamma^\sim \rho_1 \rho_2}{T_\Gamma \rho_{01} \rho_{12} : \mathbf{Tm} \Omega \Gamma^\sim \rho_0 \rho_2} \end{aligned}$$

Note that while \sim was an operation defined by induction on the syntax, \sim is a constructor of the syntax. On types, \sim is heterogeneous: it is a relation between two terms of the same type but substituted by substitutions which are related by $\Gamma \sim$. It is reflexive, symmetric and transitive and comes with coercion and coherence operators.

$$\begin{array}{c}
\frac{A : \mathbf{Ty}_j \Gamma \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma \sim \rho_0 \rho_1 \quad t_0 : \mathbf{Tm} \Omega (A[\rho_0]) \quad t_1 : \mathbf{Tm} \Omega (A[\rho_1])}{A \sim \rho_{01} t_0 t_1 : \mathbf{Tm} \Omega \mathbf{Prop}_j} \\
\\
\frac{A : \mathbf{Ty}_j \Gamma \quad t : \mathbf{Tm} \Omega (A[\rho])}{\mathbf{R}_A t : \mathbf{Tm} \Omega \underline{A \sim (\mathbf{R}_\Gamma \rho) t t}} \quad \frac{A : \mathbf{Ty}_j \Gamma \quad t_{01} : \mathbf{Tm} \Omega \underline{A \sim \rho_{01} t_0 t_1}}{\mathbf{S}_A t_{01} : \mathbf{Tm} \Omega \underline{A \sim (\mathbf{S}_\Gamma \rho_{01}) t_0 t_1}} \\
\\
\frac{A : \mathbf{Ty}_j \Gamma \quad t_{01} : \mathbf{Tm} \Omega \underline{A \sim \rho_{01} t_0 t_1} \quad t_{12} : \mathbf{Tm} \Omega \underline{A \sim \rho_{12} t_1 t_2}}{\mathbf{T}_A t_{01} t_{12} : \mathbf{Tm} \Omega \underline{A \sim (\mathbf{T}_\Gamma \rho_{01} \rho_{12}) t_0 t_2}} \\
\\
\frac{A : \mathbf{Ty}_j \Gamma \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma \sim \rho_0 \rho_1 \quad t_0 : \mathbf{Tm} \Omega A[\rho_0]}{\mathbf{coe}_A \rho_{01} t_0 : \mathbf{Tm} \Omega (A[\rho_1]) \quad \mathbf{coh}_A \rho_{01} t_0 : \mathbf{Tm} \Omega \underline{A \sim \rho_{01} t_0 (\mathbf{coe}_A \rho_{01} t_0)}}
\end{array}$$

On substitutions and terms \sim expresses congruence.

$$\frac{\delta : \mathbf{Sub} \Gamma \Delta \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma \sim \rho_0 \rho_1}{\delta \sim \rho_{01} : \mathbf{Tm} \Omega \underline{\Delta \sim (\delta \circ \rho_0) (\delta \circ \rho_1)}} \quad \frac{t : \mathbf{Tm} \Gamma A \quad \rho_{01} : \mathbf{Tm} \Omega \Gamma \sim \rho_0 \rho_1}{t \sim \rho_{01} : \mathbf{Tm} \Omega \underline{A \sim \rho_{01} (t[\rho_0]) (t[\rho_1])}}$$

We state the following definitional equalities on how the equality types and coercions compute.

$$\begin{array}{ll}
\cdot \sim \epsilon \epsilon & = \top \\
(\Gamma, x : A) \sim (\rho_0, x \mapsto t_0) (\rho_1, x \mapsto t_1) & = \sigma(\rho_{01} : \Gamma \sim \rho_0 \rho_1). A \sim \rho_{01} t_0 t_1 \\
(A[\delta]) \sim \rho_{01} t_0 t_1 & = A \sim (\delta \sim \rho_{01}) t_0 t_1 \\
(\Pi(x : A). B) \sim \rho_{01} t_0 t_1 & = \pi(x_0 : A[\rho_0]). \pi(x_1 : A[\rho_1]). \pi(x_{01} : \underline{A \sim \rho_{01} x_0 x_1}). \\
& \quad B \sim (\rho_{01}, x_{01}) (t_0 @ x_0) (t_1 @ x_1) \\
(\Sigma(x : A). B) \sim \rho_{01} (u_0, v_0) (u_1, v_1) & = \sigma(u_{01} : A \sim \rho_{01} u_0 u_1). B \sim (\rho_{01}, u_{01}) v_0 v_1 \\
\mathbf{Bool} \sim \rho_{01} t_0 t_1 & = \text{if } t_0 \text{ then (if } t_1 \text{ then } \top \text{ else } \perp) \text{ else (if } t_1 \text{ then } \perp \text{ else } \top) \\
\mathbf{Prop} \sim \rho_{01} a_0 a_1 & = (a_0 \Rightarrow a_1) \times (a_1 \Rightarrow a_0) \\
\underline{a} \sim \rho_{01} t_0 t_1 & = \top \\
\mathbf{coe}_{A[\delta]} \rho_{01} t_0 & = \mathbf{coe}_A (\delta \sim \rho_{01}) t_0 \\
\mathbf{coe}_{\Pi(x : A). B} \rho_{01} t_0 & = \lambda x_1. \mathbf{coe}_B (\rho_{01}, \mathbf{S}_A (\mathbf{coh}_A (\mathbf{S}_\Gamma \rho_{01}) x_1)) \\
& \quad (t_0 @ \mathbf{coe}_A (\mathbf{S}_\Gamma \rho_{01}) x_1) \\
\mathbf{coe}_{\Sigma(x : A). B} \rho_{01} (u_0, v_0) & = (\mathbf{coe}_A \rho_{01} u_0, \mathbf{coe}_B (\rho_{01}, \mathbf{coh}_A \rho_{01} u_0) v_0) \\
\mathbf{coe}_{\mathbf{Bool}} \rho_{01} t_0 & = t_0 \\
\mathbf{coe}_{\mathbf{Prop}} \rho_{01} a_0 & = a_0 \\
\mathbf{coe}_{\underline{a}} \rho_{01} t_0 & = \mathbf{pr}_0 (a \sim \rho_{01}) @ t_0
\end{array}$$

In addition, we have the following substitution laws.

$$\begin{aligned} (I^\sim \rho_0 \rho_1)[\nu] &= I^\sim (\rho_0 \circ \nu) (\rho_1 \circ \nu) \\ (A^\sim \rho_{01} t_0 t_1)[\nu] &= A^\sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) \\ (\text{coe}_A \rho_{01} t_0)[\nu] &= \text{coe}_A (\rho_{01}[\nu]) (t_0[\nu]) \end{aligned}$$

We only need to state these for I^\sim , A^\sim and coe_A as all the other rules coming from the translation are true by **irr**. Note that e.g. the equality for $(\Pi(x:A).B)^\sim$ is not merely a convenience, but this is the rule which adds function extensionality.

We conclude the definition of setoid type theory by adding the definitional equality for coercing along reflexivity.

$$\text{coe}_A (R_\Gamma \rho) t = t$$

Justification. The setoid translation extends to all the extra rules of setoid type theory. As all the new syntactic components are terms, we have to implement the $|-|$ and the $-^\sim$ operations for terms as specified in Section 4.2. Most components are modelled by their black counterparts because the purpose of the new rules is precisely to reflect the extra structure of the setoid translation. All the equalities are justified (T^3 is three steps transitivity, see Appendix C for all the details).

$$\begin{aligned} |I^\sim \rho_0 \rho_1| &:= I^\sim |\rho_0| |\rho_1| \\ |A^\sim \rho_{01} t_0 t_1| &:= A^\sim |\rho_{01}| |t_0| |t_1| \\ (I^\sim \rho_0 \rho_1)^\sim \tau_{01} &:= (\lambda \rho_{01}. \mathsf{T}^3_\Gamma (\mathsf{S}_\Gamma (\rho_0 \sim \tau_{01})) \rho_{01} (\rho_1 \sim \tau_{01})), \\ &\quad \lambda \rho_{01}. \mathsf{T}^3_\Gamma (\rho_0 \sim \tau_{01}) \rho_{01} (\mathsf{S}_\Gamma (\rho_1 \sim \tau_{01})) \\ (A^\sim \rho_{01} t_0 t_1)^\sim \tau_{01} &:= (\lambda t_{01}. \mathsf{T}^3_A (\mathsf{S}_A (t_0 \sim \tau_{01})) t_{01} (t_1 \sim \tau_{01})), \\ &\quad \lambda t_{01}. \mathsf{T}^3_A (t_0 \sim \tau_{01}) t_{01} (\mathsf{S}_A (t_1 \sim \tau_{01})) \end{aligned}$$

$$\begin{array}{ll} |R_\Gamma \rho| &:= R_\Gamma |\rho| & |R_A t| &:= R_A |t| \\ |\mathsf{S}_\Gamma \rho_{01}| &:= \mathsf{S}_\Gamma |\rho_{01}| & |\mathsf{S}_A t_{01}| &:= \mathsf{S}_A |t_{01}| \\ |\mathsf{T}_\Gamma \rho_{01} \rho_{12}| &:= \mathsf{T}_\Gamma |\rho_{01}| |\rho_{12}| & |\mathsf{T}_A t_{01} t_{12}| &:= \mathsf{T}_A |t_{01}| |t_{12}| \\ |\text{coe}_A \rho_{01} t_0| &:= \text{coe}_A |\rho_{01}| |t_0| & |\delta^\sim \rho_{01}| &:= \delta^\sim |\rho_{01}| \\ |\text{coh}_A \rho_{01} t_0| &:= \text{coh}_A |\rho_{01}| |t_0| & |t^\sim \rho_{01}| &:= t^\sim |\rho_{01}| \end{array}$$

Relationship to Martin-Löf's identity type (as given in Section 4.4). The rules of the traditional identity are admissible in setoid type theory. The translation is the following.

$$\begin{aligned} \text{Id}_A u v &:= A^\sim (R_\Gamma \text{id}) u v \\ \text{refl}_u &:= R_A u \\ \text{transport}_{x.P} e t &:= \text{coe}_P (R_\Gamma \text{id}, e) t \\ \text{Id}\beta t &:= \mathsf{S}_{P[x \mapsto u]} (\text{coh}_P (R_\Gamma \text{id}, R_A u) t) \\ \text{funext } e &:= \lambda x_0 x_1 x_{01}. \mathsf{T}_B (e @ x_0) (t_1 \sim (R_\Gamma \text{id}) @ x_0 @ x_1 @ x_{01}) \end{aligned}$$

$$\begin{aligned}
\text{propext } t &:= t \\
\text{Id}[] &: (\text{Id}_A u v)[\nu] = (A \sim (\mathbf{R}_\Gamma \text{id}) u v)[\nu] = \\
&\quad A \sim ((\mathbf{R}_\Gamma \text{id})[\nu]) (u[\nu]) (v[\nu]) \stackrel{\text{irr}}{=} A \sim (\nu \sim (\mathbf{R}_\Theta \text{id})) (u[\nu]) (v[\nu]) = \\
&\quad (A[\nu]) \sim (\mathbf{R}_\Theta \text{id}) (u[\nu]) (v[\nu]) = \text{Id}_{A[\nu]} (u[\nu]) (v[\nu]) \\
\text{transport}[] &: (\text{transport}_{x.P} e t)[\nu] = (\text{coe}_P (\mathbf{R}_\Gamma \text{id}, e) t)[\nu] \stackrel{\text{coe}[]^P}{=} \\
&\quad \text{coe}_P ((\mathbf{R}_\Gamma \text{id}, e)[\nu]) (t[\nu]) \stackrel{\text{irr}}{=} \text{coe}_P (\nu \sim (\mathbf{R}_\Theta \text{id}), e[\nu]) (t[\nu]) = \\
&\quad \text{coe}_{P[\nu]} (\mathbf{R}_\Theta \text{id}, e[\nu]) (t[\nu]) = \text{transport}_{x.P[\nu]} (e[\nu]) (t[\nu])
\end{aligned}$$

The other direction does not work. For example, the following definitional equalities do not hold in $\text{MLTT}_{\text{Prop}}$ extended with Martin-Löf’s identity type, however they hold in setoid type theory where **transport** is translated as above:

“constant predicate”: $\text{transport}_{x.\text{Bool}} e t = t$

“funext computes”: $\text{transport}_{f.P[y \mapsto f @ u]} (\text{funext } e) t = \text{transport}_{y.P} (e @ u) t$

As setoid type theory reflects the setoid translation, we conjecture that it is complete, that is, if $|t| = |t'|$ for any two terms $t, t' : \mathbf{Tm} \Gamma A$ of setoid type theory, then $t = t'$.

6 Conclusions and further work

We have presented a type theory which justifies both function extensionality and propositional extensionality. Compared with [1], it adds propositional extensionality and a definitional computation rule for transport, presents an equational theory and the results are checked formally. Compared with [4], it provides a translation into intensional type theory without requiring extensional type theory as a reference.

It is clear that the theory follows the setoid translation, hence we conjecture completeness with respect to this model. A corollary would be canonicity for our theory.

We expect that the translation can be extended with a universe of setoids where equality is equality of codes and quotient inductive types. Our Agda formalisation of the setoid model already supports such a universe and quotient types.

The theory is less powerful than cubical type theory [12] but the semantic justification is much more straightforward and for many practical applications, this type theory is sufficient. It also supports some definitional equalities which do not hold in cubical type theory. We believe that our programme can be extended, first of all to obtain a syntax for a groupoid type theory using our informal method to derive a theory from a translation. We also expect that we could derive an alternative explanation and implementation of homotopy type theory.

References

1. Altenkirch, T.: Extensional equality in intensional type theory. In: 14th Symposium on Logic in Computer Science. pp. 412 – 420 (1999)
2. Altenkirch, T., Kaposi, A.: Type theory in type theory using quotient inductive types. In: Bodik, R., Majumdar, R. (eds.) Proceedings of POPL 2016, St. Petersburg, FL, USA, January, 2016. pp. 18–29. ACM (2016). <https://doi.org/10.1145/2837614.2837638>
3. Altenkirch, T., Kaposi, A.: Towards a Cubical Type Theory without an Interval. In: Uustalu, T. (ed.) TYPES 2015. Leibniz International Proceedings in Informatics (LIPIcs), vol. 69, pp. 3:1–3:27. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018). <https://doi.org/10.4230/LIPIcs.TYPES.2015.3>
4. Altenkirch, T., McBride, C., Swierstra, W.: Observational equality, now! In: PLPV ’07: Proceedings of the 2007 workshop on Programming languages meets program verification. pp. 57–58. ACM (2007)
5. Anand, A., Boulrier, S., Cohen, C., Sozeau, M., Tabareau, N.: Towards certified meta-programming with typed template-coq. In: ITP 2018-9th Conference on Interactive Theorem Proving (2018)
6. Atkey, R., Ghani, N., Johann, P.: A relationally parametric model of dependent type theory. In: Jagannathan, S., Sewell, P. (eds.) Proceedings of POPL ’14, San Diego, CA, USA, January 20–21, 2014. pp. 503–516. ACM (2014)
7. Bernardy, J.P., Jansson, P., Paterson, R.: Proofs for free — parametricity for dependent types. *Journal of Functional Programming* **22**(02), 107–152 (2012). <https://doi.org/10.1017/S0956796812000056>
8. Bezem, M., Coquand, T., Huber, S.: A model of type theory in cubical sets. In: 19th International Conference on Types for Proofs and Programs (TYPES 2013). vol. 26, pp. 107–128 (2014)
9. Boulrier, S.: Extending Type Theory with Syntactic Models. PhD Thesis, École des Mines de Nantes (2018)
10. Boulrier, S., Pédrôt, P.M., Tabareau, N.: The next 700 syntactical models of type theory. In: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs. pp. 182–194. CPP 2017, ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3018610.3018620>
11. Cartmell, J.: Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic* **32**, 209–243 (1986)
12. Cohen, C., Coquand, T., Huber, S., Mörtberg, A.: Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In: Uustalu, T. (ed.) TYPES 2015. Leibniz International Proceedings in Informatics (LIPIcs), vol. 69, pp. 5:1–5:34. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2018)
13. Dybjer, P.: Internal type theory. In: Lecture Notes in Computer Science. pp. 120–134. Springer (1996)
14. Gilbert, G., Cockx, J., Sozeau, M., Tabareau, N.: Definitional Proof-Irrelevance without K. In: Proceedings of POPL 2019. Lisbon, Portugal (Jan 2019)
15. Herbelin, H.: Syntactic investigations into cubical type theory. In: Espírito Santo, J., Pinto, L. (eds.) TYPES 2018. University of Minho (2018)
16. Hofmann, M.: Conservativity of equality reflection over intensional type theory. In: TYPES 95. pp. 153–164 (1995)
17. Hofmann, M.: Extensional concepts in intensional type theory. Thesis, University of Edinburgh, Department of Computer Science (1995)

18. Hofmann, M.: Syntax and semantics of dependent types. In: *Semantics and Logics of Computation*. pp. 79–130. Cambridge University Press (1997)
19. Hofmann, M., Streicher, T.: The groupoid interpretation of type theory. In: *In Venice Festschrift*. pp. 83–111. Oxford University Press (1996)
20. Jaber, G., Lewertowski, G., Pédro, P., Sozeau, M., Tabareau, N.: The definitional side of the forcing. In: Grohe, M., Koskinen, E., Shankar, N. (eds.) *Proceedings of LICS '16*, New York, NY, USA, July 5-8, 2016. pp. 367–376. ACM (2016). <https://doi.org/10.1145/2933575.2935320>
21. Kaposi, A.: The setoid model formalized in Agda (May 2019), <https://bitbucket.org/akaposi/setoid/src/master/agda>
22. Kaposi, A., Kovács, A., Altenkirch, T.: Constructing quotient inductive-inductive types. *Proc. ACM Program. Lang.* **3**(POPL), 2:1–2:24 (Jan 2019). <https://doi.org/10.1145/3290315>
23. Martin-Löf, P.: *Intuitionistic type theory*, *Studies in Proof Theory*, vol. 1. Bibliopolis (1984)
24. Matthieu Sozeau, N.T.: Univalence for free (2013), <http://hal.inria.fr/hal-00786589/en>
25. McBride, C.: Elimination with a motive. In: *Selected Papers from the International Workshop on Types for Proofs and Programs*. pp. 197–216. TYPES '00, Springer-Verlag, Berlin, Heidelberg (2002)
26. Oury, N.: *Extensionality in the calculus of constructions*, pp. 278–293. Springer Berlin Heidelberg, Berlin, Heidelberg (2005)
27. Program, T.U.F.: *Homotopy type theory: Univalent foundations of mathematics*. Tech. rep., Institute for Advanced Study (2013)
28. Sterling, J., Angiuli, C., Gratzer, D.: Cubical syntax for reflection-free extensional equality. In: Geuvers, H. (ed.) *Proceedings of the 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019)*. vol. 131 (2019). <https://doi.org/10.4230/LIPIcs.FSCD.2019.32>
29. Streicher, T.: *Investigations into intensional type theory*. habilitation thesis (1993)
30. Tabareau, N., Tanter, É., Sozeau, M.: Equivalences for Free. In: *ICFP 2018 - International Conference on Functional Programming*. Saint-Louis, United States (Sep 2018). <https://doi.org/10.1145/3234615>
31. Van Der Walt, P., Swierstra, W.: Engineering proof by reflection in agda. In: *Symposium on Implementation and Application of Functional Languages*. pp. 157–173. Springer (2012)
32. Winterhalter, T., Sozeau, M., Tabareau, N.: Using reflection to eliminate reflection. In: Santo, J.E., Pinto, L. (eds.) *24th International Conference on Types for Proofs and Programs, TYPES 2018*. University of Minho (2018)

A Full syntax of $\mathbf{MLTT}_{\mathbf{Prop}}$

Sorts:

$$\frac{}{\mathbf{Con}_i : \mathbf{Set}} \quad \frac{\Gamma : \mathbf{Con}_i}{\mathbf{T}_j \Gamma : \mathbf{Set}} \quad \frac{\Gamma : \mathbf{Con}_i \quad \Delta : \mathbf{Con}_j}{\mathbf{Sub} \Gamma \Delta : \mathbf{Set}} \quad \frac{\Gamma : \mathbf{Con}_i \quad A : \mathbf{T}_j \Gamma}{\mathbf{Tm} \Gamma A : \mathbf{Set}}$$

Substitution calculus:

$$\begin{aligned} & \frac{}{\cdot : \mathbf{Con}_0} \quad \frac{\Gamma : \mathbf{Con}_i \quad A : \mathbf{T}_j \Gamma}{(\Gamma, x : A) : \mathbf{Con}_{i \sqcup j}} \quad \frac{A : \mathbf{T}_i \Delta \quad \delta : \mathbf{Sub} \Gamma \Delta}{A[\delta] : \mathbf{T}_i \Gamma} \\ & \frac{\Gamma : \mathbf{Con}_i}{\mathbf{id}_\Gamma : \mathbf{Sub} \Gamma \Gamma} \quad \frac{\delta : \mathbf{Sub} \Theta \Delta \quad \nu : \mathbf{Sub} \Gamma \Theta}{\delta \circ \nu : \mathbf{Sub} \Gamma \Delta} \quad \frac{\Gamma : \mathbf{Con}_i}{\epsilon_\Gamma : \mathbf{Sub} \Gamma \cdot} \\ & \frac{\delta : \mathbf{Sub} \Gamma \Delta \quad t : \mathbf{Tm} \Gamma (A[\delta])}{(\delta, x \mapsto t) : \mathbf{Sub} \Gamma (\Delta, x : A)} \quad \frac{\delta : \mathbf{Sub} \Gamma (\Delta, x : A)}{\delta : \mathbf{Sub} \Gamma \Delta} \quad \frac{\delta : \mathbf{Sub} \Gamma (\Delta, x : A)}{x[\delta] : \mathbf{Tm} \Gamma (A[\delta])} \\ & \frac{t : \mathbf{Tm} \Delta A \quad \delta : \mathbf{Sub} \Gamma \Delta}{t[\delta] : \mathbf{Tm} \Gamma (A[\delta])} \quad [\mathbf{id}] : A[\mathbf{id}] = A \quad [\circ] : A[\delta \circ \nu] = A[\delta][\nu] \\ & \mathbf{id} \circ : \mathbf{id} \circ \delta = \delta \quad \mathbf{oid} : \delta \circ \mathbf{id} = \delta \quad \circ \circ : (\delta \circ \nu) \circ \tau = \delta \circ (\nu \circ \tau) \\ & \cdot \eta : (\delta : \mathbf{Sub} \Gamma \cdot) = \epsilon \quad , \beta_0 : (\delta, x \mapsto t) = \delta \quad , \beta_1 : x[(\delta, x \mapsto t)] = t \\ & , \eta : (\delta, x \mapsto x[\delta]) = \delta \quad , \circ : (\delta, x \mapsto t) \circ \nu = (\delta \circ \nu, x \mapsto t[\nu]) \end{aligned}$$

Π types:

$$\begin{aligned} & \frac{A : \mathbf{T}_i \Gamma \quad B : \mathbf{T}_j (\Gamma, x : A)}{\Pi(x : A).B : \mathbf{T}_{i \sqcup j} \Gamma} \quad \frac{t : \mathbf{Tm} (\Gamma, x : A) B}{\lambda x. t : \mathbf{Tm} \Gamma (\Pi(x : A).B)} \\ & \frac{t : \mathbf{Tm} \Gamma (\Pi(x : A).B)}{t @ x : \mathbf{Tm} (\Gamma, x : A) B} \quad \Pi \beta : (\lambda x. t) @ x = t \quad \Pi \eta : \lambda x. t @ x = t \\ & \Pi [] : (\Pi(x : A).B)[\nu] = \Pi(x : A[\nu]).B[\nu] \quad \lambda [] : (\lambda x. t)[\nu] = \lambda x. t[\nu] \end{aligned}$$

Σ types (we write $A \times B$ for $\Sigma(x : A).B$ when x does not appear in B):

$$\begin{aligned} & \frac{A : \mathbf{T}_i \Gamma \quad B : \mathbf{T}_j (\Gamma, x : A)}{\Sigma(x : A).B : \mathbf{T}_{i \sqcup j} \Gamma} \quad \frac{u : \mathbf{Tm} \Gamma A \quad v : \mathbf{Tm} \Gamma (B[x \mapsto u])}{(u, v) : \mathbf{Tm} \Gamma (\Sigma(x : A).B)} \\ & \frac{t : \mathbf{Tm} \Gamma (\Sigma(x : A).B)}{\mathbf{pr}_0 t : \mathbf{Tm} \Gamma A} \quad \frac{t : \mathbf{Tm} \Gamma (\Sigma(x : A).B)}{\mathbf{pr}_1 t : \mathbf{Tm} \Gamma (B[x \mapsto \mathbf{pr}_0 t])} \\ & \Sigma \beta_0 : \mathbf{pr}_0 (u, v) = u \quad \Sigma \beta_1 : \mathbf{pr}_1 (u, v) = v \quad \Sigma \eta : (\mathbf{pr}_0 t, \mathbf{pr}_1 t) = t \\ & \Sigma [] : (\Sigma(x : A).B)[\nu] = \Sigma(x : A[\nu]).B[\nu] \quad , [] : (u, v)[\nu] = (u[\nu], v[\nu]) \end{aligned}$$

Booleans:

$$\frac{}{\text{Bool} : \text{Ty}_0 \Gamma} \quad \frac{}{\text{true} : \text{Tm } \Gamma \text{ Bool}} \quad \frac{}{\text{false} : \text{Tm } \Gamma \text{ Bool}}$$

$$\frac{\begin{array}{l} C : \text{Ty}_i(\Gamma, x : \text{Bool}) \\ t : \text{Tm } \Gamma \text{ Bool} \\ u : \text{Tm } \Gamma (C[x \mapsto \text{true}]) \\ v : \text{Tm } \Gamma (C[x \mapsto \text{false}]) \end{array}}{\text{if } t \text{ then } u \text{ else } v : \text{Tm } \Gamma (C[x \mapsto t])}$$

$$\begin{array}{ll} \text{Bool}\beta_{\text{true}} : \text{if true then } u \text{ else } v = u & \\ \text{Bool}\beta_{\text{false}} : \text{if false then } u \text{ else } v = v & \\ \text{Bool}[] : \text{Bool}[\nu] = \text{Bool} & \\ \text{true}[] : \text{true}[\nu] = \text{true} & \\ \text{false}[] : \text{false}[\nu] = \text{false} & \\ \text{if}[] : (\text{if } t \text{ then } u \text{ else } v)[\nu] = \text{if } t[\nu] \text{ then } u[\nu] \text{ else } v[\nu] & \end{array}$$

Propositions:

$$\frac{}{\text{Prop}_i : \text{Ty}_{i+1} \Gamma} \quad \frac{a : \text{Tm } \Gamma \text{ Prop}_i}{\underline{a} : \text{Ty}_i \Gamma} \quad \frac{u : \text{Tm } \Gamma \underline{a} \quad v : \text{Tm } \Gamma \underline{a}}{\text{irr}_a : u = v}$$

$$\frac{A : \text{Ty}_i \Gamma \quad b : \text{Tm } (\Gamma, x : A) \text{ Prop}_j}{\pi(x : A).b : \text{Tm } \Gamma \text{ Prop}_{i \sqcup j}} \quad \frac{t : \text{Tm } (\Gamma, x : A) \underline{b}}{\lambda x. t : \text{Tm } \Gamma \pi(x : A). \underline{b}}$$

$$\frac{t : \text{Tm } \Gamma \pi(x : A). \underline{b}}{t @ x : \text{Tm } (\Gamma, x : A) \underline{b}} \quad \frac{a : \text{Tm } \Gamma \text{ Prop}_i \quad b : \text{Tm } (\Gamma, x : \underline{a}) \text{ Prop}_j}{\sigma(x : a).b : \text{Tm } \Gamma \text{ Prop}_{i \sqcup j}}$$

$$\frac{u : \text{Tm } \Gamma \underline{a} \quad v : \text{Tm } \Gamma \underline{b[x \mapsto u]}}{(u, v) : \text{Tm } \Gamma \underline{\sigma(x : a).b}} \quad \frac{t : \text{Tm } \Gamma \underline{\sigma(x : a).b}}{\text{pr}_0 t : \text{Tm } \Gamma \underline{a}} \quad \frac{t : \text{Tm } \Gamma \underline{\sigma(x : a).b}}{\text{pr}_1 t : \text{Tm } \Gamma \underline{a[x \mapsto \text{pr}_0 t]}}$$

$$\frac{}{\top : \text{Tm } \Gamma \text{ Prop}_0} \quad \frac{}{\text{tt} : \text{Tm } \Gamma \perp} \quad \frac{}{\perp : \text{Tm } \Gamma \text{ Prop}_0} \quad \frac{C : \text{Ty}_i \Gamma \quad t : \text{Tm } \Gamma \perp}{\text{exfalse } t : \text{Tm } \Gamma C}$$

$$\begin{array}{ll} \text{Prop}[] : \text{Prop}_i[\nu] = \text{Prop}_i & \\ [] : \underline{a}[\nu] = \underline{a}[\nu] & \\ \pi[] : (\pi(x : A).b)[\nu] = \pi(x : A[\nu]).b[\nu] & \\ \sigma[] : (\sigma(x : a).b)[\nu] = \sigma(x : a[\nu]).b[\nu] & \\ [] : (u, v)[\nu] = (u[\nu], v[\nu]) & \\ \top[] : \top[\nu] = \top & \\ \perp[] : \perp[\nu] = \perp & \\ \text{exfalse}[] : (\text{exfalse } t)[\nu] = \text{exfalse } (t[\nu]) & \end{array}$$

B Complete implementation of the setoid translation

B.1 Specification

$$\begin{array}{c}
\Gamma : \text{Con}_i \\
\hline
|\Gamma| : \text{Con}_i \\
I^\sim : \text{Sub } \Omega \mid \Gamma \rightarrow \text{Sub } \Omega \mid \Gamma \rightarrow \text{Tm } \Omega \text{ Prop}_i \\
I^\sim [] : (I^\sim \rho_0 \rho_1)[\nu] = I^\sim (\rho_0 \circ \nu) (\rho_1 \circ \nu) \\
R_\Gamma : (\rho : \text{Sub } \Omega \mid \Gamma) \rightarrow \text{Tm } \Omega \underline{I^\sim \rho} \rho \\
S_\Gamma : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1} \rightarrow \text{Tm } \Omega \underline{I^\sim \rho_1 \rho_0} \\
T_\Gamma : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1} \rightarrow \text{Tm } \Omega \underline{I^\sim \rho_1 \rho_2} \rightarrow \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_2} \\
A : \text{Ty}_j \Gamma \\
\hline
|A| : \text{Ty}_j \Gamma \\
A^\sim : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1} \rightarrow \text{Tm } \Omega (|A|[\rho_0]) \rightarrow \text{Tm } \Omega (|A|[\rho_1]) \rightarrow \text{Tm } \Omega \text{ Prop}_j \\
A^\sim [] : (A^\sim \rho_{01} t_0 t_1)[\nu] = A^\sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) \\
R_A : (t : \text{Tm } \Omega (|A|[\rho])) \rightarrow \text{Tm } \Omega \underline{A^\sim (R_\Gamma \rho) t} t \\
S_A : \text{Tm } \Omega \underline{A^\sim \rho_{01} t_0 t_1} \rightarrow \text{Tm } \Omega \underline{A^\sim (S_\Gamma \rho_{01}) t_1 t_0} \\
T_A : \text{Tm } \Omega \underline{A^\sim \rho_{01} t_0 t_1} \rightarrow \text{Tm } \Omega \underline{A^\sim \rho_{12} t_1 t_2} \rightarrow \text{Tm } \Omega \underline{A^\sim (T_\Gamma \rho_{01} \rho_{12}) t_0 t_2} \\
\text{coe}_A : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1} \rightarrow \text{Tm } \Omega (|A|[\rho_0]) \rightarrow \text{Tm } \Omega (|A|[\rho_1]) \\
\text{coe}[]_A : (\text{coe}_A \rho_{01} t_0)[\nu] = \text{coe}_A (\rho_{01}[\nu]) (t_0[\nu]) \\
\text{coh}_A : (\rho_{01} : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1})(t_0 : \text{Tm } \Omega (|A|[\rho_0])) \rightarrow \\
\text{Tm } \Omega \underline{A^\sim \rho_{01} t_0 (\text{coe}_A \rho_{01} t_0)} \\
\delta : \text{Sub } \Gamma \Delta \\
\hline
|\delta| : \text{Sub } \Gamma \mid \Delta \\
\delta^\sim : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1} \rightarrow \text{Tm } \Omega \underline{\Delta^\sim (|\delta| \circ \rho_0) (|\delta| \circ \rho_1)} \\
t : \text{Tm } \Gamma A \\
\hline
|t| : \text{Tm } \Gamma \mid A \\
t^\sim : (\rho_{01} : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1}) \rightarrow \text{Tm } \Omega \underline{A^\sim \rho_{01} (|t|[\rho_0]) (|t|[\rho_1])}
\end{array}$$

Abbreviations The operations coe^* and coh^* are the counterparts of coe^* and coh^* in the symmetric direction. The two T^3 operations are “three steps” transitivity.

$$\begin{aligned}
& \text{coe}^*_A (\rho_{01} : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1})(t_1 : \text{Tm } \Omega (|A|[\rho_1])) : \text{Tm } \Omega (|A|[\rho_0]) \\
& \quad := \text{coe}_A (S_\Gamma \rho_{01}) t_1 \\
& \text{coh}^*_A (\rho_{01} : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1})(t_1 : \text{Tm } \Omega (|A|[\rho_1])) : \text{Tm } \Omega \underline{A^\sim \rho_{01} (\text{coe}^*_A \rho_{01} t_1) t_1} \\
& \quad := S_A (\text{coh}_A (S_\Gamma \rho_{01}) t_1) \\
& \text{T}^3_\Gamma (\rho_{01} : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_1})(\rho_{12} : \text{Tm } \Omega \underline{I^\sim \rho_1 \rho_2})(\rho_{12} : \text{Tm } \Omega \underline{I^\sim \rho_2 \rho_3}) \\
& \quad : \text{Tm } \Omega \underline{I^\sim \rho_0 \rho_3} := T_\Gamma \rho_{01} (T_\Gamma \rho_{12} \rho_{23}) \\
& \text{T}^3_A (t_{01} : \text{Tm } \Omega \underline{A^\sim \rho_{01} t_0 t_1})(t_{12} : \text{Tm } \Omega \underline{A^\sim \rho_{12} t_1 t_2})(t_{23} : \text{Tm } \Omega \underline{A^\sim \rho_{23} t_2 t_3}) \\
& \quad : \text{Tm } \Omega \underline{A^\sim (T^3_\Gamma \rho_{01} \rho_{12} \rho_{23}) t_0 t_3} := T_A t_{01} (T_A t_{12} t_{23})
\end{aligned}$$

B.2 Implementation

We implement this specification by listing what the different components of $\text{MLTT}_{\text{Prop}}$ are mapped to. We follow the order of the presentation of $\text{MLTT}_{\text{Prop}}$ in Section 2.

The $|-|$ part of the model is almost the same as the identity translation (variant ₃ in Section 3.1). The only difference is for Π types which are interpreted by a subset of all Π types: they need to also respect the relation.

Substitution calculus The $|-|$, $-\sim$, R_- , etc. components can be given one after the other as there is no interdependency for the substitution calculus. For the substitution calculus, $|-|$ is the same as the set interpretation, $-\sim$ is the same as in the graph model.

Set (identity) interpretation for the substitution calculus.

$$\begin{aligned}
|\cdot| &:= \cdot \\
|\Gamma, x : A| &:= |\Gamma|, x : |A| \\
|A[\delta]| &:= |A|[[\delta]] \\
|\text{id}_\Gamma| &:= \text{id}_{|\Gamma|} \\
|\delta \circ \nu| &:= |\delta| \circ |\nu| \\
|\epsilon_\Gamma| &:= \epsilon_\Gamma \\
|(\delta, x \mapsto t)| &:= (|\delta|, x \mapsto |t|) \\
|\delta| &:= |\delta| \\
|x[\delta]| &:= x[[\delta]] \\
|t[\delta]| &:= |t|[[\delta]] \\
|[id]| &: |A[id]| = |A|[id] \stackrel{[id]}{=} |A| \\
|[o]| &: |A[\delta \circ \nu]| = |A|[[\delta] \circ |\nu|] \stackrel{[o]}{=} |A|[[\delta]][|\nu|] = |A[\delta][\nu]| \\
|id \circ| &: |id \circ \delta| = id \circ |\delta| \stackrel{id \circ}{=} |\delta| \\
|o id| &: |\delta \circ id| = |\delta| \circ id \stackrel{o id}{=} |\delta| \\
|o o| &: |(\delta \circ \nu) \circ \tau| = (|\delta| \circ |\nu|) \circ |\tau| \stackrel{o o}{=} |\delta| \circ (|\nu| \circ |\tau|) = |\delta \circ (\nu \circ \tau)| \\
|\epsilon \eta| &: (|\delta| : \text{Sub } |\Gamma| | \cdot |) = (|\delta| : \text{Sub } |\Gamma| \cdot) \stackrel{\epsilon \eta}{=} \epsilon = |\epsilon| \\
|, \beta_0| &: |(\delta, x \mapsto t)| = (|\delta|, x \mapsto |t|) \stackrel{\beta_0}{=} |\delta| \\
|, \beta_1| &: |x[(\delta, x \mapsto t)]| = x[(|\delta|, x \mapsto |t|)] \stackrel{\beta_1}{=} |t| \\
|, \eta| &: |(\delta, x \mapsto x[\delta])| = (|\delta|, x \mapsto x[[\delta]]) \stackrel{\eta}{=} |\delta| \\
|, \circ| &: |(\delta, x \mapsto t) \circ \nu| = (|\delta|, x \mapsto |t|) \circ |\nu| \stackrel{\circ}{=} (|\delta| \circ |\nu|, x \mapsto |t|[[\nu]]) = \\
&\quad |(\delta \circ \nu, x \mapsto t[\nu])|
\end{aligned}$$

Note that $|\delta| := |\delta|$ means that implicit weakening inside $|-|$ was interpreted by implicit weakening outside the $|-|$ operation.

Logical predicates.

$$\begin{aligned}
\cdot \sim \epsilon \epsilon &:= \top \\
(\Gamma, x : A) \sim (\rho_0, x \mapsto t_0) (\rho_1, x \mapsto t_1) &:= \sigma(\rho_{01} : \Gamma \sim \rho_0 \rho_1). A \sim \rho_{01} t_0 t_1 \\
(A[\delta]) \sim \rho_{01} t_0 t_1 &:= A \sim (\delta \sim \rho_{01}) t_0 t_1 \\
\text{id}_\Gamma \sim \rho_{01} &:= \rho_{01} \\
(\delta \circ \nu) \sim \rho_{01} &:= \delta \sim (\nu \sim \rho_{01}) \\
\epsilon \sim \rho_{01} &:= \text{tt} \\
(\delta, x \mapsto t) \sim \rho_{01} &:= (\delta \sim \rho_{01}, t \sim \rho_{01}) \\
\delta \sim \rho_{01} &:= \text{pr}_0 (\delta \sim \rho_{01}) \\
(x[\delta]) \sim \rho_{01} &:= \text{pr}_1 (\delta \sim \rho_{01}) \\
(t[\delta]) \sim \rho_{01} &:= t \sim (\delta \sim \rho_{01}) \\
[\text{id}] \sim &: (A[\text{id}]) \sim \rho_{01} t_0 t_1 = A \sim (\text{id} \sim \rho_{01}) t_0 t_1 = A \sim \rho_{01} t_0 t_1 \\
[\circ] \sim &: (A[\delta \circ \nu]) \sim \rho_{01} t_0 t_1 = A \sim ((\delta \circ \nu) \sim \rho_{01}) t_0 t_1 = \\
&A \sim (\delta \sim (\nu \sim \rho_{01})) t_0 t_1 = (A[\delta][\nu]) \sim \rho_{01} t_0 t_1 \\
\text{id} \circ \sim &: (\text{id} \circ \delta) \sim \rho_{01} \stackrel{\text{irr}}{=} \delta \sim \rho_{01} \\
\circ \text{id} \sim &: (\delta \circ \text{id}) \sim \rho_{01} \stackrel{\text{irr}}{=} \delta \sim \rho_{01} \\
\circ \circ \sim &: ((\delta \circ \nu) \circ \tau) \sim \rho_{01} \stackrel{\text{irr}}{=} (\delta \circ (\nu \circ \tau)) \sim \rho_{01} \\
\epsilon \eta \sim &: \delta \sim \rho_{01} \stackrel{\text{irr}}{=} \epsilon \sim \rho_{01} \\
, \beta_0 \sim &: (\delta, x \mapsto t) \sim \rho_{01} \stackrel{\text{irr}}{=} \delta \sim \rho_{01} \\
, \beta_1 \sim &: (x[(\delta, x \mapsto t)]) \sim \rho_{01} \stackrel{\text{irr}}{=} t \sim \rho_{01} \\
, \eta \sim &: (\delta, x \mapsto x[\delta]) \sim \rho_{01} \stackrel{\text{irr}}{=} \delta \sim \rho_{01} \\
, \circ \sim &: ((\delta, x \mapsto t) \circ \nu) \sim \rho_{01} \stackrel{\text{irr}}{=} (\delta \circ \nu, x \mapsto t[\nu]) \sim \rho_{01}
\end{aligned}$$

Substitution laws for logical predicates.

$$\begin{aligned}
\cdot \sim [] &: (\cdot \sim \epsilon \epsilon)[\nu] = \top[\nu] \stackrel{\top []}{=} \top = \cdot \sim (\epsilon[\nu]) (\epsilon[\nu]) \\
(\Gamma, x : A) \sim [] &: ((\Gamma, x : A) \sim (\rho_0, x \mapsto t_0) (\rho_1, x \mapsto t_1))[\nu] = \\
&\sigma(\rho_{01} : (\Gamma \sim \rho_0 \rho_1)[\nu]). (A \sim \rho_{01} t_0 t_1)[\nu] \stackrel{\Gamma \sim [], A \sim []}{=} \\
&\sigma(\rho_{01} : \Gamma \sim (\rho_0 \circ \nu) (\rho_1 \circ \nu)). A \sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) = \\
&(\Gamma, x : A) \sim (\rho_0 \circ \nu, x \mapsto t_0[\nu]) (\rho_1 \circ \nu, x \mapsto t_1[\nu]) \stackrel{\circ}{=} \\
&(\Gamma, x : A) \sim ((\rho_0, x \mapsto t_0) \circ \nu) ((\rho_1, x \mapsto t_1) \circ \nu) \\
(A[\delta]) \sim [] &: ((A[\delta]) \sim \rho_{01} t_0 t_1)[\nu] = (A \sim (\delta \sim \rho_{01}) t_0 t_1)[\nu] \stackrel{A \sim []}{=} \\
&A \sim ((\delta \sim \rho_{01})[\nu]) (t_0[\nu]) (t_1[\nu]) \stackrel{\text{irr}}{=} \\
&A \sim (\delta \sim (\rho_{01}[\nu])) (t_0[\nu]) (t_1[\nu]) =
\end{aligned}$$

$$(A[\delta])^\sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu])$$

Reflexivity, symmetry and transitivity.

$$\begin{array}{ll}
\mathbf{R}, \epsilon & := \mathbf{tt} \\
\mathbf{R}_\Gamma, x : A (\rho, x \mapsto t) & := (\mathbf{R}_\Gamma \rho, \mathbf{R}_A t) \\
\mathbf{R}_{A[\delta]} t & := \mathbf{R}_A t \\
\mathbf{R}_{[\text{id}]} & : \mathbf{R}_{A[\text{id}]} t = \mathbf{R}_A t \\
\mathbf{R}_{[\circ]} & : \mathbf{R}_{A[\delta \circ \nu]} t = \mathbf{R}_A t = \mathbf{R}_{A[\delta][\nu]} t \\
\mathbf{S}, \mathbf{tt} & := \mathbf{tt} \\
\mathbf{S}_\Gamma, x : A (\rho_{01}, t_{01}) & := (\mathbf{S}_\Gamma \rho_{01}, \mathbf{S}_A t_{01}) \\
\mathbf{S}_{A[\delta]} t_{01} & := \mathbf{S}_A t_{01} \\
\mathbf{S}_{[\text{id}]} & : \mathbf{S}_{A[\text{id}]} t_{01} = \mathbf{S}_A t_{01} \\
\mathbf{S}_{[\circ]} & : \mathbf{S}_{A[\delta \circ \nu]} t_{01} = \mathbf{S}_A t_{01} = \mathbf{S}_{A[\delta][\nu]} t_{01} \\
\mathbf{T}, \mathbf{tt} \mathbf{tt} & := \mathbf{tt} \\
\mathbf{T}_\Gamma, x : A (\rho_{01}, t_{01}) (\rho_{12}, t_{12}) & := (\mathbf{T}_\Gamma \rho_{01} \rho_{12}, \mathbf{T}_A t_{01} t_{12}) \\
\mathbf{T}_{A[\delta]} t_{01} t_{12} & := \mathbf{T}_A t_{01} t_{12} \\
\mathbf{T}_{[\text{id}]} & : \mathbf{T}_{A[\text{id}]} t_{01} t_{12} = \mathbf{T}_A t_{01} t_{12} \\
\mathbf{T}_{[\circ]} & : \mathbf{T}_{A[\delta \circ \nu]} t_{01} t_{12} = \mathbf{T}_A t_{01} t_{12} = \mathbf{T}_{A[\delta][\nu]} t_{01} t_{12}
\end{array}$$

Coercion and coherence.

$$\begin{array}{ll}
\text{coe}_{A[\delta]} \rho_{01} t_0 & := \text{coe}_A (\delta^\sim \rho_{01}) t_0 \\
\text{coe}_{[\text{id}]} & : \text{coe}_{A[\text{id}]} \rho_{01} t_0 = \text{coe}_A (\text{id}^\sim \rho_{01}) t_0 = \text{coe}_A \rho_{01} t_0 \\
\text{coe}_{[\circ]} & : \text{coe}_{A[\delta \circ \nu]} \rho_{01} t_0 = \text{coe}_A (\delta^\sim (\nu^\sim \rho_{01})) t_0 = \text{coe}_{A[\delta][\nu]} \rho_{01} t_0 \\
\text{coe}_{[\]_{A[\delta]}} & : (\text{coe}_{A[\delta]} \rho_{01} t_0)[\nu] = (\text{coe}_A (\delta^\sim \rho_{01}) t_0)[\nu] \stackrel{\text{coe}_{[\]_A}}{=} \\
& \quad \text{coe}_A ((\delta^\sim \rho_{01})[\nu]) (t_0[\nu]) \stackrel{\text{irr}}{=} \text{coe}_A (\delta^\sim (\rho_{01}[\nu])) (t_0[\nu]) = \text{coe}_{A[\]} \\
\text{coh}_{A[\delta]} \rho_{01} t_0 & := \text{coh}_A (\delta^\sim \rho_{01}) t_0 \\
\text{coh}_{[\text{id}]} & : \text{coh}_{A[\text{id}]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \text{coh}_A \rho_{01} t_0 \\
\text{coh}_{[\circ]} & : \text{coh}_{A[\delta \circ \nu]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \text{coh}_{A[\delta][\nu]} \rho_{01} t_0
\end{array}$$

Π

$$\begin{array}{ll}
|\Pi(x : A).B| & := \Sigma(f : \Pi(x : |A|).|B|). \pi(x_0 : |A|). \pi(x_1 : |A|). \\
& \quad \frac{\pi(x_{01} : A^\sim (\mathbf{R}_\Gamma \text{id}) x_0 x_1). B^\sim (\mathbf{R}_\Gamma \text{id}, x_{01}) (f @ x_0) (f @ x_1)}{} \\
(\Pi(x : A).B)^\sim \rho_{01} t_0 t_1 & := \pi(x_0 : |A|[\rho_0]). \pi(x_1 : |A|[\rho_1]). \pi(x_{01} : A^\sim \rho_{01} x_0 x_1). \\
& \quad B^\sim (\rho_{01}, x_{01}) (\mathbf{pr}_0 t_0 @ x_0) (\mathbf{pr}_0 t_1 @ x_1) \\
(\Pi(x : A).B)^\sim [\] & : ((\Pi(x : A).B)^\sim \rho_{01} t_0 t_1)[\nu] =
\end{array}$$

$$\begin{aligned}
& \pi(x_0 : A[\rho_0 \circ \nu]).\pi(x_1 : A[\rho_1 \circ \nu]).\pi(x_{01} : \underline{A^\sim(\rho_{01}[\nu]) x_0 x_1}). \\
& B^\sim(\rho_{01}[\nu], x_{01})(\text{pr}_0(t_0[\nu]) @ x_0)(\text{pr}_0(t_1[\nu]) @ x_1) = \\
& (\Pi(x : A).B)^\sim(\rho_{01}[\nu])(t_0[\nu])(t_1[\nu]) \\
R_{\Pi(x : A).B} t & := \text{pr}_1 t \\
S_{\Pi(x : A).B} t_{01} & := \lambda x_0 x_1 x_{01}. S_B(t_{01} @ x_1 @ x_0 @ S_A x_{01}) \\
T_{\Pi(x : A).B} t_{01} t_{12} & := \\
& \lambda x_0 x_2 x_{02}. T_B(t_{01} @ x_0 @ \text{coe}_A \rho_{01} x_0 @ \text{coh}_A \rho_{01} x_0) \\
& (t_{12} @ \text{coe}_A \rho_{01} x_0 @ x_2 @ T_A(S_A(\text{coh}_A \rho_{01} x_0)) x_{02}) \\
\text{coe}_{\Pi(x : A).B} \rho_{01} t_0 & := \left(\lambda x_1. \text{coe}_B(\rho_{01}, \text{coh}^*_A \rho_{01} x_1)(\text{pr}_0 t_0 @ \text{coe}^*_A \rho_{01} x_1), \right. \\
& \lambda x_1 x_2 x_{12}. \left(T^3_B(S_B(\text{coh}_B(\rho_{01}, x_{10})(\text{pr}_0 t_0 @ x_0))) \right. \\
& (\text{pr}_1 t_0 @ x_0 @ x_3 @ T^3_A(S_A x_{10}) x_{12} x_{23}) \\
& (\text{coh}_B(\rho_{01}, x_{23})(\text{pr}_0 t_0 @ x_3))) \\
& \left. [x_0 \mapsto \text{coe}^*_A \rho_{01} x_1, x_{10} \mapsto \text{coh}^*_A \rho_{01} x_1, \right. \\
& \left. x_3 \mapsto \text{coe}^*_A \rho_{01} x_2, x_{23} \mapsto \text{coh}^*_A \rho_{01} x_2] \right) \\
\text{coe}[]_{\Pi(x : A).B} & : \\
& (\text{coe}_{\Pi(x : A).B} \rho_{01} t_0)[\nu] = \\
& (\lambda x_1. (\text{coe}_B(\rho_{01}, \text{coh}^*_A \rho_{01} x_1)(\text{pr}_0 t_0 @ \text{coe}^*_A \rho_{01} x_1))[\nu], \dots) \stackrel{\text{coe}[]_B}{=} \\
& (\lambda x_1. \text{coe}_B(\rho_{01}[\nu], (\text{coh}^*_A \rho_{01} x_1)[\nu])(\text{pr}_0(t_0[\nu]) @ (\text{coe}^*_A \rho_{01} x_1)[\nu]), \dots) \stackrel{\text{coe}[]_A}{=} \\
& (\lambda x_1. \text{coe}_B(\rho_{01}[\nu], (\text{coh}^*_A(\rho_{01}[\nu]) x_1))(\text{pr}_0(t_0[\nu]) @ \text{coe}^*_A(\rho_{01}[\nu]) x_1), \dots) \stackrel{\text{irr}}{=} \\
& \text{coe}_{\Pi(x : A).B}(\rho_{01}[\nu])(t_0[\nu]) \\
\text{coh}_{\Pi(x : A).B} \rho_{01} t_0 & := \\
& \lambda x_0 x_1 x_{01}. T_B(\text{pr}_1 t_0 @ x_0 @ \text{coe}^*_A \rho_{01} x_1 @ T_A x_{01}(S_A(\text{coh}^*_A \rho_{01} x_1))) \\
& (\text{coh}_B(\rho_{01}, \text{coh}^*_A \rho_{01} x_1)(\text{pr}_0 t_0 @ \text{coe}^*_A \rho_{01} x_1)) \\
|\lambda x. t| & := (\lambda x. |t|, \lambda x_0 x_1 x_{01}. t^\sim(\text{R}_\Gamma \text{id}, x_{01})) \\
(\lambda x. t)^\sim \rho_{01} & := \lambda x_0 x_1 x_{01}. t^\sim(\rho_{01}, x_{01}) \\
|t @ x| & := \text{pr}_0 |t| @ x \\
(t @ x)^\sim(\rho_{01}, x_{01}) & := t^\sim \rho_{01} @ x_0 @ x_1 @ x_{01} \\
|\Pi\beta| & : |(\lambda x. t) @ x| = \text{pr}_0 |\lambda x. t| @ x = (\lambda x. |t|) @ x \stackrel{\Pi\beta}{=} |t| \\
|\Pi\eta| & : |\lambda x. t @ x| = (\lambda x. |t @ x|, \lambda x_0 x_1 x_{01}. (t @ x)^\sim(\text{R}_\Gamma \text{id}, x_{01})) = \\
& (\lambda x. \text{pr}_0 |t| @ x, \lambda x_0 x_1 x_{01}. t^\sim(\text{R}_\Gamma \text{id}) @ x_0 @ x_1 @ x_{01}) \stackrel{\Pi\eta}{=} \\
& (\text{pr}_0 |t|, t^\sim(\text{R}_\Gamma \text{id})) \stackrel{\text{irr}}{=} (\text{pr}_0 |t|, \text{pr}_1 |t|) \stackrel{\Sigma\eta}{=} |t|
\end{aligned}$$

$$\begin{aligned}
\Pi\beta^\sim & : ((\lambda x.t) @ x)^\sim \rho_{01} \stackrel{\text{irr}}{=} t^\sim \rho_{01} \\
\Pi\eta^\sim & : (\lambda x.t @ x)^\sim \rho_{01} \stackrel{\text{irr}}{=} t^\sim \rho_{01} \\
|\Pi[]| & : \\
|\Pi(x:A).B|[\nu]| & = |\Pi(x:A).B|[\nu]| \stackrel{\Pi[], \pi[], A^\sim[], B^\sim[]}{=} \\
\Sigma(f : \Pi(x : |A|[\nu]|).|B|[\nu]|). \pi(x_0 : |A|[\nu]|). \pi(x_1 : |A|[\nu]|). & \\
\pi(x_{01} : A^\sim((R_\Gamma \text{id})[\nu]) x_0 x_1). B^\sim((R_\Gamma \text{id})[\nu], x_{01})(f @ x_0)(f @ x_1) & \stackrel{\text{irr}}{=} \\
|\Pi(x : A[\nu]).B|[\nu]| & \\
\Pi[]^\sim & : \\
((\Pi(x:A).B)|[\nu])^\sim \rho_{01} t_0 t_1 & = (\Pi(x:A).B)^\sim (\nu^\sim \rho_{01}) t_0 t_1 = \\
\pi(x_0 : A[\nu \circ \rho_0]). \pi(x_1 : A[\nu \circ \rho_1]). \pi(x_{01} : A^\sim(\nu^\sim \rho_{01}) x_0 x_1). & \\
B^\sim(\nu^\sim \rho_{01}, x_{01})(\text{pr}_0 t_0 @ x_0)(\text{pr}_0 t_1 @ x_1) & = (\Pi(x:A[\nu]).B|[\nu])^\sim \rho_{01} t_0 t_1 \\
R_{\Pi[]} & : R_{(\Pi(x:A).B)|[\nu]} t \stackrel{\text{irr}}{=} R_{\Pi(x:A[\nu]).B|[\nu]} t \\
S_{\Pi[]} & : S_{(\Pi(x:A).B)|[\nu]} t_{01} \stackrel{\text{irr}}{=} S_{\Pi(x:A[\nu]).B|[\nu]} t_{01} \\
T_{\Pi[]} & : T_{(\Pi(x:A).B)|[\nu]} t_{01} t_{12} \stackrel{\text{irr}}{=} T_{\Pi(x:A[\nu]).B|[\nu]} t_{01} t_{12} \\
\text{coe}_{\Pi[]} & : \\
\text{coe}_{(\Pi(x:A).B)|[\nu]} \rho_{01} t_0 & = \text{coe}_{\Pi(x:A).B} (\nu^\sim \rho_{01}) t_0 = \\
(\lambda x_1. \text{coe}_B(\nu^\sim \rho_{01}, \text{coh}^*_A(\nu^\sim \rho_{01}) x_1)(\text{pr}_0 t_0 @ \text{coe}^*_A(\nu^\sim \rho_{01}) x_1), \dots) & \stackrel{\text{irr}}{=} \\
(\lambda x_1. \text{coe}_{B|[\nu]}(\rho_{01}, \text{coh}^*_{A[\nu]} \rho_{01} x_1)(\text{pr}_0 t_0 @ \text{coe}^*_{A[\nu]} \rho_{01} x_1), \dots) & = \\
\text{coe}_{\Pi(x:A[\nu]).B|[\nu]} \rho_{01} t_0 & \\
\text{coh}_{\Pi[]} & : \text{coh}_{(\Pi(x:A).B)|[\nu]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \text{coh}_{\Pi(x:A[\nu]).B|[\nu]} \rho_{01} t_0 \\
|\lambda[]| & : |(\lambda x.t)|[\nu]| = |\lambda x.t|[\nu]| \stackrel{\lambda[]}{=} \lambda x. |t|[\nu]| = |\lambda x.t|[\nu]| \\
\lambda[]^\sim & : ((\lambda x.t)|[\nu])^\sim \rho_{01} \stackrel{\text{irr}}{=} (\lambda x.t|[\nu])^\sim \rho_{01} \\
\Sigma & \\
|\Sigma(x:A).B| & := \Sigma(x : |A|).|B| \\
(\Sigma(x:A).B)^\sim \rho_{01} (u_0, v_0) (u_1, v_1) & := \sigma(x_{01} : A^\sim \rho_{01} u_0 u_1). B^\sim(\rho_{01}, x_{01}) v_0 v_1 \\
(\Sigma(x:A).B)^\sim [] & : \\
((\Sigma(x:A).B)^\sim \rho_{01} (u_0, v_0) (u_1, v_1))[\nu] & = \\
\sigma(x_{01} : A^\sim \rho_{01} u_0 u_1)[\nu]. (B^\sim(\rho_{01}, x_{01}) v_0 v_1)[\nu] & \stackrel{A^\sim[], B^\sim[]}{=} \\
\sigma(x_{01} : A^\sim(\rho_{01}[\nu]) (u_0[\nu]) (u_1[\nu])). B^\sim(\rho_{01}[\nu], x_{01}) (v_0[\nu]) (v_1[\nu]) & = \\
(\Sigma(x:A).B)^\sim(\rho_{01}[\nu]) ((u_0, v_0)[\nu]) ((u_1, v_1)[\nu]) & \\
R_{\Sigma(x:A).B} (u, v) & := (R_A u, R_B v) \\
S_{\Sigma(x:A).B} (u_{01}, v_{01}) & := (S_A u_{01}, S_B v_{01})
\end{aligned}$$

$$\begin{aligned}
\mathsf{T}_{\Sigma(x:A).B} (u_{01}, v_{01}) (u_{12}, v_{12}) &:= (\mathsf{T}_A u_{01} u_{12}, \mathsf{Tm}_B v_{01} v_{12}) \\
\mathsf{coe}_{\Sigma(x:A).B} \rho_{01} (u_0, v_0) &:= (\mathsf{coe}_A \rho_{01} u_0, \mathsf{coe}_B (\rho_{01}, \mathsf{coh}_A \rho_{01} u_0) v_0) \\
\mathsf{coe}[]_{\Sigma(x:A).B} &: \\
&(\mathsf{coe}_{\Sigma(x:A).B} \rho_{01} (u_0, v_0))[\nu] = \\
&((\mathsf{coe}_A \rho_{01} u_0)[\nu], (\mathsf{coe}_B (\rho_{01}, \mathsf{coh}_A \rho_{01} u_0) v_0)[\nu]) \stackrel{\mathsf{coe}[]_A, \mathsf{coe}[]_B}{=} \\
&(\mathsf{coe}_A (\rho_{01}[\nu]) (u_0[\nu]), \mathsf{coe}_B (\rho_{01}[\nu], \mathsf{coh}_A (\rho_{01}[\nu]) (u_0[\nu])) (v_0[\nu])) = \\
&\mathsf{coe}_{\Sigma(x:A).B} (\rho_{01}[\nu]) ((u_0, v_0)[\nu]) \\
\mathsf{coh}_{\Sigma(x:A).B} \rho_{01} (u_0, v_0) &:= (\mathsf{coh}_A \rho_{01} u_0, \mathsf{coh}_B (\rho_{01}, \mathsf{coh}_A \rho_{01} u_0) v_0) \\
|(u, v)| &:= (|u|, |v|) \\
(u, v) \sim \rho_{01} &:= (u \sim \rho_{01}, v \sim \rho_{01}) \\
|\mathsf{pr}_0 t| &:= \mathsf{pr}_0 |t| \\
(\mathsf{pr}_0 t) \sim \rho_{01} &:= \mathsf{pr}_0 (t \sim \rho_{01}) \\
|\mathsf{pr}_1 t| &:= \mathsf{pr}_1 |t| \\
(\mathsf{pr}_1 t) \sim \rho_{01} &:= \mathsf{pr}_1 (t \sim \rho_{01}) \\
|\Sigma\beta_0| &: |\mathsf{pr}_0 (u, v)| = \mathsf{pr}_0 (|u|, |v|) \stackrel{\Sigma\beta_0}{=} |u| \\
\Sigma\beta_0 \sim &: (\mathsf{pr}_0 (u, v)) \sim \rho_{01} \stackrel{\text{irr}}{=} u \sim \rho_{01} \\
|\Sigma\beta_1| &: |\mathsf{pr}_1 (u, v)| = \mathsf{pr}_1 (|u|, |v|) \stackrel{\Sigma\beta_1}{=} |v| \\
\Sigma\beta_1 \sim &: (\mathsf{pr}_1 (u, v)) \sim \rho_{01} \stackrel{\text{irr}}{=} v \sim \rho_{01} \\
|\Sigma\eta| &: |(\mathsf{pr}_0 t, \mathsf{pr}_1 t)| = (\mathsf{pr}_0 |t|, \mathsf{pr}_1 |t|) \stackrel{\Sigma\eta}{=} |t| \\
\Sigma\eta \sim &: (\mathsf{pr}_0 t, \mathsf{pr}_1 t) \sim \rho_{01} \stackrel{\text{irr}}{=} t \sim \rho_{01} \\
|\Sigma[]| &: |(\Sigma(x:A).B)[\nu]| = |\Sigma(x:A).B|[\nu] \stackrel{\Sigma[]}{=} \\
&\Sigma(x: |A|[\nu]).|B|[\nu] = |\Sigma(x:A[\nu]).B[\nu]| \\
\Sigma[] \sim &: ((\Sigma(x:A).B)[\nu]) \sim \rho_{01} (u_0, v_0) (u_1, v_1) = \\
&(\Sigma(x:A).B) \sim (\nu \sim \rho_{01}) (u_0, v_0) (u_1, v_1) = \\
&\sigma(x_{01}: A \sim (\nu \sim \rho_{01}) u_0 u_1).B \sim (\nu \sim \rho_{01}, x_{01}) v_0 v_1 = \\
&(\Sigma(x:A[\nu]).B[\nu]) \sim \rho_{01} (u_0, v_0) (u_1, v_1) \\
\mathsf{R}_{\Sigma[]} &: \mathsf{R}_{(\Sigma(x:A).B)[\nu]} t \stackrel{\text{irr}}{=} \mathsf{R}_{\Sigma(x:A[\nu]).B[\nu]} t \\
\mathsf{S}_{\Sigma[]} &: \mathsf{S}_{(\Sigma(x:A).B)[\nu]} t_{01} \stackrel{\text{irr}}{=} \mathsf{S}_{\Sigma(x:A[\nu]).B[\nu]} t_{01} \\
\mathsf{T}_{\Sigma[]} &: \mathsf{T}_{(\Sigma(x:A).B)[\nu]} t_{01} t_{12} \stackrel{\text{irr}}{=} \mathsf{T}_{\Sigma(x:A[\nu]).B[\nu]} t_{01} t_{12} \\
\mathsf{coe}_{\Sigma[]} &: \mathsf{coe}_{(\Sigma(x:A).B)[\nu]} \rho_{01} (u_0, v_0) = \\
&\mathsf{coe}_{\Sigma(x:A).B} (\nu \sim \rho_{01}) (u_0, v_0) = \\
&(\mathsf{coe}_A (\nu \sim \rho_{01}) u_0, \mathsf{coe}_B (\nu \sim \rho_{01}, \mathsf{coh}_A (\nu \sim \rho_{01}) u_0) v_0) = \\
&(\mathsf{coe}_{A[\nu]} \rho_{01} u_0, \mathsf{coe}_{B[\nu]} (\rho_{01}, \mathsf{coh}_{A[\nu]} \rho_{01} u_0) v_0) =
\end{aligned}$$

$$\begin{aligned}
& \text{coe}_{\Sigma(x:A[\nu]).B[\nu]} \rho_{01} (u_0, v_0) \\
\text{coh}_{\Sigma\Box} & : \text{coh}_{(\Sigma(x:A).B)[\nu]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \text{coh}_{\Sigma(x:A[\nu]).B[\nu]} \rho_{01} t_0 \\
|, \Box| & : |(u, v)[\nu]| = |(u, v)|[\Box] = (|u|, |v|)[\Box] \stackrel{\Box}{=} \\
& (|u|[\Box], |v|[\Box]) = |(u[\nu], v[\nu])| \\
, \Box^\sim & : ((u, v)[\nu])^\sim \rho_{01} \stackrel{\text{irr}}{=} (u[\nu], v[\nu])^\sim \rho_{01}
\end{aligned}$$

Bool

$$\begin{aligned}
|\text{Bool}| & := \text{Bool} \\
\text{Bool}^\sim \rho_{01} t_0 t_1 & := \text{if } t_0 \text{ then } (\text{if } t_1 \text{ then } \top \text{ else } \perp) \text{ else } (\text{if } t_1 \text{ then } \perp \text{ else } \top) \\
\text{Bool}^\sim \Box & : \\
& (\text{Bool}^\sim \rho_{01} t_0 t_1)[\nu] = \\
& \text{if } t_0[\nu] \text{ then } (\text{if } t_1[\nu] \text{ then } \top \text{ else } \perp) \text{ else } (\text{if } t_1[\nu] \text{ then } \perp \text{ else } \top) = \\
& \text{Bool}^\sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) \\
R_{\text{Bool}} t & := \text{if } t \text{ then tt else tt} \\
S_{\text{Bool}} t_{01} & := \text{if } t_0 \text{ then } (\text{if } t_1 \text{ then tt else exfalse } t_{01}) \\
& \quad \text{else } (\text{if } t_1 \text{ then exfalse } t_{01} \text{ else tt}) \\
T_{\text{Bool}} t_{01} t_{12} & := \text{if } t_0 \text{ then } (\text{if } t_1 \text{ then } (\text{if } t_2 \text{ then tt else exfalse } t_{12}) \text{ else exfalse } t_{01}) \\
& \quad \text{else } (\text{if } t_1 \text{ then exfalse } t_{01} \text{ else } (\text{if } t_2 \text{ then exfalse } t_{12} \text{ else tt})) \\
\text{coe}_{\text{Bool}} \rho_{01} t_0 & := t_0 \\
\text{coe}_{\Box}^{\text{Bool}} & : (\text{coe}_{\text{Bool}} \rho_{01} t_0)[\nu] = t_0[\nu] = \text{coe}_{\text{Bool}} \rho_{01} (t_0[\nu]) \\
\text{coh}_{\text{Bool}} \rho_{01} t_0 & := \text{if } t_0 \text{ then tt else tt} \\
|\text{true}| & := \text{true} \\
\text{true}^\sim \rho_{01} & := \text{tt} \\
|\text{false}| & := \text{false} \\
\text{false}^\sim \rho_{01} & := \text{tt} \\
|\text{if } t \text{ then } u \text{ else } v| & := \text{if } |t| \text{ then } |u| \text{ else } |v| \\
(\text{if } t \text{ then } u \text{ else } v)^\sim \rho_{01} & := \text{if } t[\rho_0] \text{ then } (\text{if } t[\rho_1] \text{ then } u^\sim \rho_{01} \text{ else exfalse } (u^\sim \rho_{01})) \\
& \quad \text{else } (\text{if } t[\rho_1] \text{ then exfalse } (v^\sim \rho_{01}) \text{ else } v^\sim \rho_{01}) \\
|\text{Bool}^{\beta_{\text{true}}}| & : |\text{if true then } u \text{ else } v| = \text{if true then } |u| \text{ else } |v| \stackrel{\text{Bool}^{\beta_{\text{true}}}}{=} |u| \\
|\text{Bool}^{\beta_{\text{false}}}| & : |\text{if false then } u \text{ else } v| = \text{if false then } |u| \text{ else } |v| \stackrel{\text{Bool}^{\beta_{\text{false}}}}{=} |v| \\
|\text{Bool}[\Box]| & : |\text{Bool}[\nu]| = |\text{Bool}|[\Box] \stackrel{\text{Bool}[\Box]}{=} \text{Bool} = |\text{Bool}| \\
\text{Bool}[\Box]^\sim & : (\text{Bool}[\nu])^\sim \rho_{01} t_0 t_1 = \text{Bool}^\sim (\nu^\sim \rho_{01}) t_0 t_1 = \\
& \text{if } t_0 \text{ then } (\text{if } t_1 \text{ then } \top \text{ else } \perp) \text{ else } (\text{if } t_1 \text{ then } \perp \text{ else } \top) = \\
& \text{Bool}^\sim \rho_{01} t_0 t_1
\end{aligned}$$

$$\begin{aligned}
\mathbf{R}_{\mathbf{Bool}[]} &: \mathbf{R}_{\mathbf{Bool}[\nu]} t \stackrel{\text{irr}}{=} \mathbf{R}_{\mathbf{Bool}} t \\
\mathbf{S}_{\mathbf{Bool}[]} &: \mathbf{S}_{\mathbf{Bool}[\nu]} t_{01} \stackrel{\text{irr}}{=} \mathbf{S}_{\mathbf{Bool}} t_{01} \\
\mathbf{T}_{\mathbf{Bool}[]} &: \mathbf{T}_{\mathbf{Bool}[\nu]} t_{01} t_{12} \stackrel{\text{irr}}{=} \mathbf{T}_{\mathbf{Bool}} t_{01} t_{12} \\
\mathbf{coe}_{\mathbf{Bool}[]} &: \mathbf{coe}_{\mathbf{Bool}[\nu]} \rho_{01} t_0 = \mathbf{coe}_{\mathbf{Bool}} (\nu \sim \rho_{01}) t_0 = t_0 = \mathbf{coe}_{\mathbf{Bool}} \rho_{01} t_0 \\
\mathbf{coh}_{\mathbf{Bool}[]} &: \mathbf{coh}_{\mathbf{Bool}[\nu]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \mathbf{coh}_{\mathbf{Bool}} \rho_{01} t_0 \\
|\mathbf{true}[]| &: |\mathbf{true}[\nu]| = \mathbf{true}[\nu] \stackrel{\text{true}[]}{=} \mathbf{true} = |\mathbf{true}| \\
\mathbf{true}[] \sim &: (\mathbf{true}[\nu]) \sim \rho_{01} \stackrel{\text{irr}}{=} \mathbf{true} \sim \rho_{01} \\
|\mathbf{false}[]| &: |\mathbf{false}[\nu]| = \mathbf{false}[\nu] \stackrel{\text{false}[]}{=} \mathbf{false} = |\mathbf{false}| \\
\mathbf{false}[] \sim &: (\mathbf{false}[\nu]) \sim \rho_{01} \stackrel{\text{irr}}{=} \mathbf{false} \sim \rho_{01} \\
|\mathbf{if}[]| &: |(\mathbf{if} t \text{ then } u \text{ else } v)[\nu]| = (\mathbf{if} |t| \text{ then } |u| \text{ else } |v|)[\nu] \stackrel{\text{if}[]}{=} \\
&\quad \mathbf{if} |t|[\nu] \text{ then } |u|[\nu] \text{ else } |v|[\nu] = |\mathbf{if} t[\nu] \text{ then } u[\nu] \text{ else } v[\nu]| \\
\mathbf{if}[] \sim &: ((\mathbf{if} t \text{ then } u \text{ else } v)[\nu]) \sim \rho_{01} \stackrel{\text{irr}}{=} (\mathbf{if} t[\nu] \text{ then } u[\nu] \text{ else } v[\nu]) \sim \rho_{01}
\end{aligned}$$

Prop

$$\begin{aligned}
|\mathbf{Prop}_i| &:= \mathbf{Prop}_i \\
\mathbf{Prop}_i \sim \rho_{01} a_0 a_1 &:= (\underline{a_0} \Rightarrow a_1) \times (\underline{a_1} \Rightarrow a_0) \\
\mathbf{Prop}_i \sim [] &: (\mathbf{Prop}_i \sim \rho_{01} a_0 a_1)[\nu] \stackrel{\pi[], \sigma[], []}{=} \\
&\quad (\underline{a_0}[\nu] \Rightarrow a_1[\nu]) \times (\underline{a_1}[\nu] \Rightarrow a_0[\nu]) = \\
&\quad \mathbf{Prop}_i \sim (\rho_{01}[\nu]) (a_0[\nu]) (a_1[\nu]) \\
\mathbf{R}_{\mathbf{Prop}_i} a &:= (\lambda x.x, \lambda x.x) \\
\mathbf{S}_{\mathbf{Prop}_i} (a_{01}, a_{10}) &:= (a_{10}, a_{01}) \\
\mathbf{T}_{\mathbf{Prop}_i} (a_{01}, a_{10}) (a_{12}, a_{21}) &:= (\lambda x_0.x_{12} @ (a_{01} @ x_0), \lambda x_2.x_{10} @ (a_{21} @ x_2)) \\
\mathbf{coe}_{\mathbf{Prop}_i} \rho_{01} a_0 &:= a_0 \\
\mathbf{coe}[]_{\mathbf{Prop}_i} &: (\mathbf{coe}_{\mathbf{Prop}_i} \rho_{01} a_0)[\nu] = a_0[\nu] = \mathbf{coe}_{\mathbf{Prop}_i} \rho_{01} (a_0[\nu]) \\
\mathbf{coh}_{\mathbf{Prop}_i} \rho_{01} a_0 &:= (\lambda x.x, \lambda x.x) \\
|\underline{a}| &:= |\underline{a}| \\
\underline{a} \sim \rho_{01} t_0 t_1 &:= \top \\
\underline{a} \sim [] &: (\underline{a} \sim \rho_{01} t_0 t_1)[\nu] = \top[\nu] \stackrel{\top[]}{=} \top = \underline{a} \sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu]) \\
\mathbf{R}_{\underline{a}} t &:= \mathbf{tt} \\
\mathbf{S}_{\underline{a}} t_{01} &:= \mathbf{tt} \\
\mathbf{T}_{\underline{a}} t_{01} t_{12} &:= \mathbf{tt} \\
\mathbf{coe}_{\underline{a}} \rho_{01} t_0 &:= \mathbf{pr}_0 (\underline{a} \sim \rho_{01}) @ t_0
\end{aligned}$$

$$\begin{aligned}
\text{coe}[\underline{a}] &: (\text{coe}_{\underline{a}} \rho_{01} t_0)[\nu] = \text{pr}_0((a \sim \rho_{01})[\nu]) @ t_0[\nu] \stackrel{\text{irr}}{=} \\
&\quad \text{pr}_0(a \sim (\rho_{01}[\nu])) @ t_0[\nu] = \text{coe}_{\underline{a}}(\rho_{01}[\nu])(t_0[\nu]) \\
\text{coh}_{\underline{a}} \rho_{01} t_0 &:= \text{tt} \\
|\text{irr}_a| &: |u : \mathbf{Tm} \Gamma \underline{a}| = (|u| : \mathbf{Tm} |\Gamma| |\underline{a}|) \stackrel{\text{irr}_{|\underline{a}|}}{=} (|v| : \mathbf{Tm} |\Gamma| |\underline{a}|) = \\
&\quad |v : \mathbf{Tm} \Gamma \underline{a}| \\
|\pi(x : A).b| &:= \pi(x : |A|).|b| \\
(\pi(x : A).b) \sim \rho_{01} &:= (\lambda f_0 x_1. \text{pr}_0(b \sim (\rho_{01}, \text{coh}^*_A \rho_{01} x_1)) @ (f_0 @ \text{coe}^*_A \rho_{01} x_1), \\
&\quad \lambda f_1 x_0. \text{pr}_1(b \sim (\rho_{01}, \text{coh}_A \rho_{01} x_0)) @ (f_1 @ \text{coe}_A \rho_{01} x_0)) \\
|\lambda x. t| &:= \lambda x. |t| \\
(\lambda x. t) \sim \rho_{01} &:= \text{tt} \\
|t @ x| &:= |t| @ x \\
(t @ x) \sim \rho_{01} &:= \text{tt} \\
|\sigma(x : a).b| &:= \sigma(x : |a|).|b| \\
(\sigma(x : a).b) \sim \rho_{01} &:= (\lambda z_0. (\text{pr}_0(a \sim \rho_{01}) @ \text{pr}_0 z_0, \text{pr}_0(b \sim (\rho_{01}, \text{tt})) @ \text{pr}_1 z_0), \\
&\quad \lambda z_1. (\text{pr}_1(a \sim \rho_{01}) @ \text{pr}_0 z_1, \text{pr}_1(b \sim (\rho_{01}, \text{tt})) @ \text{pr}_1 z_1)) \\
|(u, v)| &:= (|u|, |v|) \\
(u, v) \sim \rho_{01} &:= \text{tt} \\
|\text{pr}_0 t| &:= \text{pr}_0 |t| \\
(\text{pr}_0 t) \sim \rho_{01} &:= \text{tt} \\
|\text{pr}_1 t| &:= \text{pr}_1 |t| \\
(\text{pr}_1 t) \sim \rho_{01} &:= \text{tt} \\
|\top| &:= \top \\
\top \sim \rho_{01} &:= (\lambda x. x, \lambda x. x) \\
|\text{tt}| &:= \text{tt} \\
\text{tt} \sim \rho_{01} &:= \text{tt} \\
|\perp| &:= \perp \\
\perp \sim \rho_{01} &:= (\lambda x. x, \lambda x. x) \\
|\text{exfalso } t| &:= \text{exfalso } |t| \\
(\text{exfalso } t) \sim \rho_{01} &:= \text{exfalso } (|t|[\rho_0]) \\
|\text{Prop}[]| &: |\text{Prop}[\nu]| = |\text{Prop}[][\nu]| \stackrel{\text{Prop}[]}{=} \text{Prop} = |\text{Prop}| \\
\text{Prop}[] \sim &: (\text{Prop}[\nu]) \sim \rho_{01} a_0 a_1 = \text{Prop} \sim (\nu \sim \rho_{01}) a_0 a_1 = \\
&\quad (\underline{a}_0 \Rightarrow a_1) \times (\underline{a}_1 \Rightarrow a_0) = \text{Prop} \sim \rho_{01} a_0 a_1 \\
\text{R}_{\text{Prop}[]} &: \text{R}_{\text{Prop}[\nu]} a \stackrel{\text{irr}}{=} \text{R}_{\text{Prop}} a \\
\text{S}_{\text{Prop}[]} &: \text{S}_{\text{Prop}[\nu]} a_{01} \stackrel{\text{irr}}{=} \text{S}_{\text{Prop}} a_{01}
\end{aligned}$$

$\mathsf{T}_{\mathsf{Prop}}[]$:	$\mathsf{T}_{\mathsf{Prop}[\nu]} a_{01} a_{12} \stackrel{\text{irr}}{=} \mathsf{T}_{\mathsf{Prop}} a_{01} a_{12}$
$\mathsf{coe}_{\mathsf{Prop}}[]$:	$\mathsf{coe}_{\mathsf{Prop}[\nu]} \rho_{01} a_0 = \mathsf{coe}_{\mathsf{Prop}} (\nu \sim \rho_{01}) a_0 = a_0 = \mathsf{coe}_{\mathsf{Prop}} \rho_{01} a_0$
$\mathsf{coh}_{\mathsf{Prop}}[]$:	$\mathsf{coh}_{\mathsf{Prop}[\nu]} \rho_{01} a_0 \stackrel{\text{irr}}{=} \mathsf{coh}_{\mathsf{Prop}} \rho_{01} a_0$
$ \underline{a} $:	$ \underline{a}[\nu] = \underline{a} [\nu] \stackrel{\underline{a}[]}{=} \underline{a} [\nu] = \underline{a}[\nu] $
$\underline{a} \sim$:	$(\underline{a}[\nu]) \sim \rho_{01} a_0 a_1 = \underline{a} \sim (\nu \sim \rho_{01}) a_0 a_1 = \top = \underline{a}[\nu] \sim \rho_{01} a_0 a_1$
$\mathsf{R}_{\underline{a}}$:	$\mathsf{R}_{\underline{a}[\nu]} t \stackrel{\text{irr}}{=} \mathsf{R}_{\underline{a}[\nu]} t$
$\mathsf{S}_{\underline{a}}$:	$\mathsf{S}_{\underline{a}[\nu]} t_{01} \stackrel{\text{irr}}{=} \mathsf{S}_{\underline{a}[\nu]} t_{01}$
$\mathsf{T}_{\underline{a}}$:	$\mathsf{T}_{\underline{a}[\nu]} t_{01} t_{12} \stackrel{\text{irr}}{=} \mathsf{T}_{\underline{a}[\nu]} t_{01} t_{12}$
$\mathsf{coe}_{\underline{a}}$:	$\mathsf{coe}_{\underline{a}[\nu]} \rho_{01} t_0 = \mathsf{coe}_{\underline{a}} (\nu \sim \rho_{01}) t_0 = \mathsf{pr}_0 (a \sim (\nu \sim \rho_{01})) @ t_0 =$ $\mathsf{pr}_0 ((a[\nu]) \sim \rho_{01}) @ t_0 = \mathsf{coe}_{\underline{a}[\nu]} \rho_{01} t_0$
$\mathsf{coh}_{\underline{a}}$:	$\mathsf{coh}_{\underline{a}[\nu]} \rho_{01} t_0 \stackrel{\text{irr}}{=} \mathsf{coh}_{\underline{a}[\nu]} \rho_{01} t_0$
$ \pi[] $:	$ (\pi(x : A).b)[\nu] = (\pi(x : A). b)[\nu] \stackrel{\pi[]}{=} \pi(x : A[\nu]). b $ $\pi(x : A[\nu]). b = \pi(x : A[\nu]).b[\nu] $
$\pi[] \sim$:	$((\pi(x : A).b)[\nu]) \sim \rho_{01} \stackrel{\text{irr}}{=} (\pi(x : A[\nu]).b[\nu]) \sim \rho_{01}$
$ \sigma[] $:	$ (\sigma(x : a).b)[\nu] = (\sigma(x : a). b)[\nu] \stackrel{\sigma[]}{=} \sigma(x : a[\nu]). b $ $\sigma(x : a[\nu]). b = \sigma(x : a[\nu]).b[\nu] $
$\sigma[] \sim$:	$((\sigma(x : a).b)[\nu]) \sim \rho_{01} \stackrel{\text{irr}}{=} (\sigma(x : a[\nu]).b[\nu]) \sim \rho_{01}$
$ \top[] $:	$ \top[\nu] = \top [\nu] \stackrel{\top[]}{=} \top = \top $
$\top[] \sim$:	$(\top[\nu]) \sim \rho_{01} = \top \sim (\nu \sim \rho_{01}) = (\lambda x.x, \lambda x.x) = \top \sim \rho_{01}$
$ \perp[] $:	$ \perp[\nu] = \perp [\nu] \stackrel{\perp[]}{=} \perp = \perp $
$\perp[] \sim$:	$(\perp[\nu]) \sim \rho_{01} = \perp \sim (\nu \sim \rho_{01}) = (\lambda x.x, \lambda x.x) = \perp \sim \rho_{01}$
$ \mathsf{exfalso}[] $:	$ (\mathsf{exfalso} t)[\nu] = (\mathsf{exfalso} t)[\nu] \stackrel{\mathsf{exfalso}[]}{=} \mathsf{exfalso} (t [\nu]) =$ $ \mathsf{exfalso} (t[\nu]) $
$\mathsf{exfalso}[] \sim$:	$((\mathsf{exfalso} t)[\nu]) \sim \rho_{01} = (\mathsf{exfalso} t) \sim (\nu \sim \rho_{01}) =$ $\mathsf{exfalso} (t [\nu] \circ \rho_0) =$ $\mathsf{exfalso} (t[\nu] [\rho_0]) = (\mathsf{exfalso} (t[\nu]) \sim \rho_{01})$

B.3 Extensions

Identity type

$ \mathsf{Id}_A u v $	$:= A \sim (\mathsf{R}_\Gamma \mathsf{id}) u v $
$(\mathsf{Id}_A t_0 t_1) \sim \rho_{01}$	$:= (\lambda x_{01}. \top^3_A (\mathsf{S}_A (t_0 \sim \rho_{01})) x_{01} (t_1 \sim \rho_{01})),$

$$\begin{aligned}
& \lambda x_{01}. \mathsf{T}^3_A (t_0 \sim \rho_{01}) x_{01} (\mathsf{S}_A (t_1 \sim \rho_{01})) \\
|\mathsf{refl}_u| & := \mathsf{R}_A |u| \\
\mathsf{refl}_t \sim \rho_{01} & := \mathsf{tt} \\
|\mathsf{transport}_{x.P} e t| & := \mathsf{coe}_P (\mathsf{R}_\Gamma \mathsf{id}, |e|) |t| \\
(\mathsf{transport}_{x.P} e t) \sim \rho_{01} & := \mathsf{T}^3_P (\mathsf{S}_P (\mathsf{coh}_P ((\mathsf{R}_\Gamma \mathsf{id}, |e|)[\rho_0]) (|t|[\rho_0]))) (t \sim \rho_{01}) \\
& \quad (\mathsf{coh}_P ((\mathsf{R}_\Gamma \mathsf{id}, |e|)[\rho_1]) (|t|[\rho_1])) \\
|\mathsf{Id}\beta| & := \mathsf{S}_{P[x \mapsto |u|]} (\mathsf{coh}_P (\mathsf{R}_\Gamma \mathsf{id}, \mathsf{R}_A |u|) |t|) \\
(\mathsf{Id}\beta t) \sim \rho_{01} & := \mathsf{tt} \\
|\mathsf{funext} e| & := \lambda x_0 x_1 x_{01}. \mathsf{T}_B (|e| @ x_0) (t_1 \sim (\mathsf{R}_\Gamma \mathsf{id}) @ x_0 @ x_1 @ x_{01}) \\
(\mathsf{funext} e) \sim \rho_{01} & := \mathsf{tt} \\
|\mathsf{propext} t| & := |t| \\
(\mathsf{propext} t) \sim \rho_{01} & := \mathsf{tt} \\
|\mathsf{Id}[]| & : \quad |(\mathsf{Id}_A u v)[\nu]| = (A \sim (\mathsf{R}_\Gamma \mathsf{id}) |u| |v|)[\nu] \stackrel{A \sim []}{=} \\
& \quad A \sim ((\mathsf{R}_\Gamma \mathsf{id})[\nu]) (|u|[\nu]) (|v|[\nu]) \stackrel{\mathsf{irr}}{=} \\
& \quad A \sim (\nu \sim (\mathsf{R}_\Theta \mathsf{id})) (|u|[\nu]) (|v|[\nu]) = \\
& \quad (A[\nu]) \sim (\mathsf{R}_\Theta \mathsf{id}) (|u|[\nu]) (|v|[\nu]) = |\mathsf{Id}_{A[\nu]} (u[\nu]) (v[\nu])| \\
|\mathsf{Id}[] \sim & : \quad ((\mathsf{Id}_A u v)[\nu]) \sim \rho_{01} \stackrel{\mathsf{irr}}{=} (\mathsf{Id}_{A[\nu]} (u[\nu]) (v[\nu])) \sim \rho_{01} \\
|\mathsf{transport}[]| & : \quad |(\mathsf{transport}_{x.P} e t)[\nu]| = (\mathsf{coe}_P (\mathsf{R}_\Gamma \mathsf{id}, |e|) |t|)[\nu] \stackrel{\mathsf{coe}[]_P}{=} \\
& \quad \mathsf{coe}_P ((\mathsf{R}_\Gamma \mathsf{id}, |e|)[\nu]) (|t|[\nu]) \stackrel{\mathsf{irr}}{=} \\
& \quad \mathsf{coe}_P (\nu \sim (\mathsf{R}_\Theta \mathsf{id}), |e|[\nu]) (|t|[\nu]) = \\
& \quad \mathsf{coe}_{P[\nu]} (\mathsf{R}_\Theta \mathsf{id}, |e|[\nu]) (|t|[\nu]) = \\
& \quad |\mathsf{transport}_{x.P[\nu]} (e[\nu]) (t[\nu])| \\
\mathsf{transport}[] \sim & : \quad ((\mathsf{transport}_{x.P} e t)[\nu]) \sim \rho_{01}
\end{aligned}$$

Definitional computation rule

$$\begin{aligned}
\mathsf{coeR}_{A[\delta]} \rho t & : \mathsf{coe}_{A[\delta]} (\mathsf{R}_\Gamma \rho) t = \mathsf{coe}_A (\delta \sim (\mathsf{R}_\Gamma \rho)) t \stackrel{\mathsf{irr}}{=} \\
& \quad \mathsf{coe}_A (\mathsf{R}_\Delta (|\delta| \circ \rho)) t \stackrel{\mathsf{coeR}_A (|\delta| \circ \rho)}{=} t \\
\mathsf{coeR}_{\Pi(x:A).B} \rho t & : \\
& \quad \mathsf{coe}_{\Pi(x:A).B} (\mathsf{R}_\Gamma \rho) t = \\
& \quad (\lambda x. \mathsf{coe}_B (\mathsf{R}_\Gamma \rho, \mathsf{coh}^*_A (\mathsf{R}_\Gamma \rho) x) (\mathsf{pr}_0 t @ \mathsf{coe}^*_A (\mathsf{R}_\Gamma \rho) x), \dots) = \\
& \quad (\lambda x. \mathsf{coe}_B (\mathsf{R}_\Gamma \rho, \mathsf{S}_A (\mathsf{coh}_A (\mathsf{S}_\Gamma (\mathsf{R}_\Gamma \rho)) x)) (\mathsf{pr}_0 t @ \mathsf{coe}_A (\mathsf{S}_\Gamma (\mathsf{R}_\Gamma \rho)) x), \dots) \stackrel{\mathsf{irr}}{=} \\
& \quad (\lambda x. \mathsf{coe}_B (\mathsf{R}_\Gamma \rho, \mathsf{S}_A (\mathsf{coh}_A (\mathsf{R}_\Gamma \rho) x)) (\mathsf{pr}_0 t @ \mathsf{coe}_A (\mathsf{R}_\Gamma \rho) x), \dots) \stackrel{\mathsf{coeR}_A \rho x}{=} \\
& \quad (\lambda x. \mathsf{coe}_B (\mathsf{R}_\Gamma \rho, \mathsf{S}_A (\mathsf{coh}_A (\mathsf{R}_\Gamma \rho) x)) (\mathsf{pr}_0 t @ x), \dots) \stackrel{\mathsf{irr}}{=}
\end{aligned}$$

$$\begin{aligned}
& (\lambda x. \text{coe}_B (R_\Gamma, x : A (\rho, x \mapsto x)) (\text{pr}_0 t @ x), \dots) \stackrel{\text{coeR}_B (\rho, x \mapsto x)}{=} (\text{pr}_0 t @ x) \\
& (\lambda x. \text{pr}_0 t @ x, \dots) \stackrel{\text{II}\eta}{=} (\text{pr}_0 t, \dots) \stackrel{\text{irr}}{=} (\text{pr}_0 t, \dots) = t \\
\text{coeR}_{\Sigma(x:A).B} \rho (u, v) : \text{coe}_{\Sigma(x:A).B} (R_\Gamma \rho) (u, v) = & \\
& (\text{coe}_A (R_\Gamma \rho) u, \text{coe}_B (R_\Gamma \rho, \text{coh}_A (R_\Gamma \rho) u) v) \stackrel{\text{coeR}_A \rho u}{=} \\
& (u, \text{coe}_B (R_\Gamma \rho, \text{coh}_A (R_\Gamma \rho) u) v) \stackrel{\text{irr}}{=} \\
& (u, \text{coe}_B (R_\Gamma, x : A (\rho, x \mapsto u)) v) \stackrel{\text{coeR}_B (\rho, x \mapsto u)}{=} (u, v) \\
\text{coeR}_{\text{Bool}} \rho t & : \text{coe}_{\text{Bool}} (R_\Gamma \rho) t = t \\
\text{coeR}_{\text{Prop}} \rho a & : \text{coe}_{\text{Prop}} (R_\Gamma \rho) a = a \\
\text{coeR}_{\underline{a}} \rho t & : \text{coe}_{\underline{a}} (R_\Gamma \rho) t = \text{pr}_0 (a \sim (R_\Gamma \rho)) @ t \stackrel{\text{irr}_{a[\rho]}}{=} t
\end{aligned}$$

C Justification of the rules of setoid type theory

The setoid model justifies all the extra rules of setoid type theory. As all the new syntactic components are terms, we have to implement the $|-|$ and the \sim operations for terms as specified in Section 4.2. Most components are modelled by their black counterparts.

$$\begin{aligned}
|I^\sim \rho_0 \rho_1| & := I^\sim |\rho_0| |\rho_1| \\
(I^\sim \rho_0 \rho_1)^\sim \tau_{01} & := (\lambda \rho_{01}. \text{T}^3_\Gamma (\text{S}_\Gamma (\rho_0 \sim \tau_{01})) \rho_{01} (\rho_1 \sim \tau_{01}), \\
& \quad \lambda \rho_{01}. \text{T}^3_\Gamma (\rho_0 \sim \tau_{01}) \rho_{01} (\text{S}_\Gamma (\rho_1 \sim \tau_{01}))) \\
|R_\Gamma \rho| & := R_\Gamma |\rho| \\
(R_\Gamma \rho)^\sim \tau_{01} & := \text{tt} \\
|S_\Gamma \rho_{01}| & := S_\Gamma |\rho_{01}| \\
(S_\Gamma \rho_{01})^\sim \tau_{01} & := \text{tt} \\
|T_\Gamma \rho_{01} \rho_{12}| & := T_\Gamma |\rho_{01}| |\rho_{12}| \\
(T_\Gamma \rho_{01} \rho_{12})^\sim \gamma_{01} & := \text{tt} \\
|A^\sim \rho_{01} t_0 t_1| & := A^\sim |\rho_{01}| |t_0| |t_1| \\
(A^\sim \rho_{01} t_0 t_1)^\sim \tau_{01} & := (\lambda t_{01}. \text{T}^3_A (\text{S}_A (t_0 \sim \tau_{01})) t_{01} (t_1 \sim \tau_{01}), \\
& \quad \lambda t_{01}. \text{T}^3_A (t_0 \sim \tau_{01}) t_{01} (\text{S}_A (t_1 \sim \tau_{01}))) \\
|R_A t| & := R_A |t| \\
(R_A t)^\sim \tau_{01} & := \text{tt} \\
|S_A t_{01}| & := S_A |t_{01}| \\
(S_A t_{01})^\sim \tau_{01} & := \text{tt} \\
|T_A t_{01} t_{12}| & := T_A |t_{01}| |t_{12}| \\
(T_A t_{01} t_{12})^\sim \tau_{01} & := \text{tt} \\
|\text{coe}_A \rho_{01} t_0| & := \text{coe}_A |\rho_{01}| |t_0|
\end{aligned}$$

$$\begin{aligned}
(\text{coe}_A \rho_{01} t_0)^\sim \tau_{01} &:= \mathsf{T}^3_A \left(\mathsf{S}_A (\text{coh}_A (|\rho_{01}| \tau_0) (|t_0| \tau_0)) \right) (t_0^\sim \tau_{01}) \\
&\quad (\text{coh}_A (|\rho_{01}| \tau_1) (|t_0| \tau_1)) \\
|\text{coh}_A \rho_{01} t_0| &:= \text{coh}_A |\rho_{01}| |t_0| \\
(\text{coh}_A \rho_{01} t_0)^\sim \tau_{01} &:= \text{tt} \\
|\delta^\sim \rho_{01}| &:= \delta^\sim |\rho_{01}| \\
(\delta^\sim \rho_{01})^\sim \tau_{01} &:= \text{tt} \\
|t^\sim \rho_{01}| &:= t^\sim |\rho_{01}| \\
(t^\sim \rho_{01})^\sim \tau_{01} &:= \text{tt}
\end{aligned}$$

All the equalities are justified. Here we only list how the $|-|$ part of the translation justifies the equalities, $-\sim$ justifies everything automatically by **irr**, as all the new syntax are terms and $-\sim$ on a term returns a proof of a proposition.

- $|\cdot^\sim \epsilon \epsilon| = \cdot^\sim |\epsilon| |\epsilon| = \top = |\top|$
- $|(\Gamma, x : A)^\sim (\rho_0, x \mapsto t_0) (\rho_1, x \mapsto t_1)| =$
 $(\Gamma, x : A)^\sim | (\rho_0, x \mapsto t_0) | | (\rho_1, x \mapsto t_1) | =$
 $(\Gamma, x : A)^\sim (|\rho_0|, x \mapsto |t_0|) (|\rho_1|, x \mapsto |t_1|) =$
 $\sigma(\rho_{01} : \Gamma^\sim |\rho_0| |\rho_1|). A^\sim \rho_{01} |t_0| |t_1| =$
 $|\sigma(\rho_{01} : \Gamma^\sim \rho_0 \rho_1). A^\sim \rho_{01} t_0 t_1|$
- $|(A[\delta])^\sim \rho_{01} t_0 t_1| = (A[\delta])^\sim |\rho_{01}| |t_0| |t_1| = A^\sim (\delta^\sim |\rho_{01}|) |t_0| |t_1| =$
 $|A^\sim (\delta^\sim \rho_{01}) t_0 t_1|$
- $|(\Pi(x : A). B)^\sim \rho_{01} t_0 t_1| = (\Pi(x : A). B)^\sim |\rho_{01}| |t_0| |t_1| =$
 $\pi(x_0 : |A| [| \rho_0 |]). \pi(x_1 : |A| [| \rho_1 |]). \pi(x_{01} : \underline{A^\sim |\rho_{01}| x_0 x_1}).$
 $B^\sim (|\rho_{01}|, x_{01}) (\text{pr}_0 |t_0| @ x_0) (\text{pr}_0 |t_1| @ x_1) =$
 $\pi(x_0 : |A| [| \rho_0 |]). \pi(x_1 : |A| [| \rho_1 |]). \pi(x_{01} : \underline{A^\sim |\rho_{01}| x_0 x_1}).$
 $B^\sim (|\rho_{01}|, x_{01}) |t_0 @ x_0| |t_1 @ x_1| =$
 $|\pi(x_0 : A[\rho_0]). \pi(x_1 : A[\rho_1]). \pi(x_{01} : \underline{A^\sim \rho_{01} x_0 x_1}).$
 $B^\sim (\rho_{01}, x_{01}) (t_0 @ x_0) (t_1 @ x_1)|$
- $|(\Sigma(x : A). B)^\sim \rho_{01} (u_0, v_0) (u_1, v_1)| =$
 $(\Sigma(x : A). B)^\sim |\rho_{01}| (|u_0|, |v_0|) (|u_1|, |v_1|) =$
 $\sigma(x_{01} : A^\sim |\rho_{01}| |u_0| |u_1|). B^\sim (|\rho_{01}|, x_{01}) |v_0| |v_1| =$
 $|\sigma(u_{01} : A^\sim \rho_{01} u_0 u_1). B^\sim (\rho_{01}, u_{01}) v_0 v_1|$
- $|\text{Bool}^\sim \rho_{01} t_0 t_1| = \text{Bool}^\sim |\rho_{01}| |t_0| |t_1| =$
 $\text{if } |t_0| \text{ then } (\text{if } |t_1| \text{ then } \top \text{ else } \perp) \text{ else } (\text{if } |t_1| \text{ then } \perp \text{ else } \top) =$
 $|\text{if } t_0 \text{ then } (\text{if } t_1 \text{ then } \top \text{ else } \perp) \text{ else } (\text{if } t_1 \text{ then } \perp \text{ else } \top)|$
- $|\text{Prop}^\sim \rho_{01} a_0 a_1| = \text{Prop}^\sim |\rho_{01}| |a_0| |a_1| = (|\underline{a_0}| \Rightarrow |a_1|) \times (|\underline{a_1}| \Rightarrow |a_0|) =$
 $|(a_0 \Rightarrow a_1) \times (a_1 \Rightarrow a_0)|$
- $|\underline{a}^\sim \rho_{01} t_0 t_1| = \underline{a}^\sim |\rho_{01}| |t_0| |t_1| = \top = |\top|$

- $|\text{coe}_{A[\delta]} \rho_{01} t_0| = \text{coe}_{A[\delta]} |\rho_{01}| |t_0| = \text{coe}_A (\delta^\sim |\rho_{01}|) |t_0| = |\text{coe}_A (\delta^\sim \rho_{01}) t_0|$
- $|\text{coe}_{\Pi(x:A).B} \rho_{01} t_0| = \text{coe}_{\Pi(x:A).B} |\rho_{01}| |t_0| =$
 $(\lambda x_1. \text{coe}_B (|\rho_{01}|, \text{coh}^*_A |\rho_{01}| x_1) (\text{pr}_0 |t_0| @ \text{coe}^*_A |\rho_{01}| x_1), \dots) =$
 $(\lambda x_1. \text{coe}_B (|\rho_{01}|, \text{coh}^*_A |\rho_{01}| x_1) (t_0 @ \text{coe}^*_A \rho_{01} x_1), \dots) =$
 $|\lambda x_1. \text{coe}_B (\rho_{01}, \text{coh}^*_A \rho_{01} x_1) (t_0 @ \text{coe}^*_A \rho_{01} x_1)|$
- $|\text{coe}_{\Sigma(x:A).B} \rho_{01} (u_0, v_0)| = \text{coe}_{\Sigma(x:A).B} |\rho_{01}| (|u_0|, |v_0|) =$
 $(\text{coe}_A |\rho_{01}| |u_0|, \text{coe}_B (|\rho_{01}|, \text{coh}_A |\rho_{01}| |u_0|) |v_0|) =$
 $|\text{coe}_A \rho_{01} u_0, \text{coe}_B (\rho_{01}, \text{coh}_A \rho_{01} u_0) v_0|$
- $|\text{coe}_{\text{Bool}} \rho_{01} t_0| = \text{coe}_{\text{Bool}} |\rho_{01}| |t_0| = |t_0|$
- $|\text{coe}_{\text{Prop}_i} \rho_{01} a_0| = \text{coe}_{\text{Prop}_i} |\rho_{01}| |a_0| = |a_0|$
- $|\text{coe}_{\underline{a}} \rho_{01} t_0| = \text{coe}_{\underline{a}} |\rho_{01}| |t_0| = \text{pr}_0 (a^\sim |\rho_{01}|) @ |t_0| = |\text{pr}_0 (a^\sim \rho_{01}) @ t_0|$
- $|(I^\sim \rho_0 \rho_1)[\nu]| = (I^\sim |\rho_0| |\rho_1|)[[\nu]] \stackrel{I^\sim \square}{=} I^\sim (|\rho_0| \circ |\nu|) (|\rho_1| \circ |\nu|) =$
 $|I^\sim (\rho_0 \circ \nu) (\rho_1 \circ \nu)|$
- $|(A^\sim \rho_{01} t_0 t_1)[\nu]| = (A^\sim |\rho_{01}| |t_0| |t_1|)[[\nu]] \stackrel{A^\sim \square}{=} A^\sim (|\rho_{01}|[[\nu]]) (|t_0|[[\nu]]) (|t_1|[[\nu]]) = |A^\sim (\rho_{01}[\nu]) (t_0[\nu]) (t_1[\nu])|$
- $|\text{coe}_A \rho_{01} t_0[\nu]| = (\text{coe}_A |\rho_{01}| |t_0|)[[\nu]] \stackrel{\text{coe} \square^A}{=} \text{coe}_A (|\rho_{01}|[[\nu]]) (|t_0|[[\nu]]) =$
 $|\text{coe}_A (\rho_{01}[\nu]) (t_0[\nu])|$
- $|\text{coe}_A (\mathbf{R}_\Gamma \rho) t| = \text{coe}_A (\mathbf{R}_\Gamma |\rho|) |t| \stackrel{\text{coe} \mathbf{R}^A}{=} |t|$